



# THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

# **Exploring Participative Learner Modelling and Its Effects on Learner Behaviour**

**Rafael Morales Gamboa**

Ph.D. Thesis  
University of Edinburgh  
2000

## Abstract

The educational benefits of involving learners as active players in the learner modelling process have been an important motivation for research on this form of learner modelling, henceforth referred to as *participative learner modelling*. Such benefits, conceived as the promotion of learners' reflection on and awareness of their own knowledge, have in most cases been asserted on the grounds of system design and supported only by anecdotal evidence.

This dissertation explores the issue of whether participative learner modelling actually promotes learners' reflection and awareness. It does so by firstly interpreting 'reflection' and 'awareness' in light of "classical" theories of human cognitive architecture, skill acquisition and meta-cognition, in order to infer changes in learner abilities (and therefore behaviour) amenable to empirical corroboration. The occurrence of such changes is then tested for an implementation of a paradigmatic form of participative learner modelling: allowing learners to inspect and modify their learner models.

The domain of application centres on the sensorimotor skill of controlling a pole on a cart and represents a novel type of domain for participative learner modelling. Special attention is paid to evaluating the method developed for constructing learner models and the form of presenting them to learners: the former is based on a method known as *behavioural cloning* for acquiring expert knowledge by means of machine learning; the latter deals with the modularity of the learner models and the modality and interactivity of their presentation.

The outcome of this research suggests that participative learner modelling may increase the abilities of learners to report accurately their problem-solving knowledge and to carry out novel tasks in the same domain—the sort of behavioural changes expected from increased learners' awareness and reflection. More importantly perhaps, the research suggests a viable methodology for examining the educational benefits of participative learner modelling. It also exemplifies the difficulties that such endeavours will face.

*A Patti, con todo mi amor*

## Acknowledgements

This is the end of a walk that started several years ago. Many people have helped me along the way, but the space is short and my memory poor . . .

First of all, I would like to thank my supervisor Helen Pain, for all her support, encouragement and most of all, friendship. I have learnt so much from her.

Thanks to my second supervisor Tom Conlon, who always had an encouraging phrase to start a list of deep comments and most helpful suggestions.

William Cohen deserves special thanks for allowing free use of RIPPER, and for his prompt and kind responses to all my questions. I would like to extend this acknowledgement to all the people who have helped to develop the many tools I was able to use for free.

Thanks to the folks in room F11, especially those in the AI in Education and AI in Music groups, for the friendly, merry, noisy and otherwise deliciously distracting environment that they have created up there. I had the fortune to share this room for a while with Shari Trewin, who filled the room with her smile and noble Scottish heart, and Mike Ramscar, who was full of ideas to drop into any attentive ear.

Back in Mexico, I would like to thank my “boss” Luis Pineda for his friendship and support. He is a great example for me to follow, even if he is not always right . . .

I am most grateful to my extended family in Mexico, especially my parents Nicolasa and Rafael, and my mother in law Leonila, whose love is so great that it crosses the ocean.

Patti: there are no words to thank you, but maybe the castles . . . Dinorah: you don’t know how much you have helped me—page 12 will be always smiling at you. Rafael: some day you’ll be able to read this and smile too.

Finally, I would like to thank the former Department of Artificial Intelligence, University of Edinburgh, for the opportunity given to conduct this research and the facilities provided to that end. The research was supported by the Instituto de Investigaciones Eléctricas and CONACyT, Mexico, under scholarship 64999/111091. I am most grateful to them.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Declaration</b>	<b>v</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Contents</b>	<b>xi</b>
<b>List of tables</b>	<b>xv</b>
<b>List of figures</b>	<b>xvii</b>
<b>Introduction and overview</b>	<b>1</b>
Overview of the dissertation . . . . .	4
<b>1 Participative learner modelling</b>	<b>7</b>
1.1 Learner modelling . . . . .	7
1.2 External influences . . . . .	9
1.3 Objectives . . . . .	10
1.4 A rough classification of approaches . . . . .	11
1.4.1 Participative learner diagnosis . . . . .	12
1.4.2 Participative model maintenance . . . . .	13
1.4.3 Participative model exploitation . . . . .	14
1.5 Research issues . . . . .	14
1.6 Participative learner modelling as educational activity . . . . .	17
1.7 How to build a bridge to Utopia . . . . .	19

<b>2</b>	<b>Effects of participative learner modelling on the learner</b>	<b>21</b>
2.1	The human cognitive architecture in a nutshell . . . . .	21
2.1.1	Skill acquisition . . . . .	24
2.1.2	Meta-cognition . . . . .	25
2.2	Effects of participative learner modelling . . . . .	26
2.3	Dependency on the theory . . . . .	28
<b>3</b>	<b>A testing domain: balancing a pole on a cart</b>	<b>31</b>
3.1	The pole and cart problem . . . . .	31
3.2	Why balancing balancing a pole on a cart? . . . . .	32
3.3	Balancing the pole and cart in PACMOD . . . . .	34
3.4	Human control of the pole and cart . . . . .	37
<b>4</b>	<b>Construction of learner models</b>	<b>41</b>
4.1	Options for learner modelling . . . . .	41
4.1.1	Knowledge-based vs interaction-driven . . . . .	42
4.1.2	Behavioural vs epistemic . . . . .	43
4.1.3	Learner modelling for the pole and cart . . . . .	43
4.2	Machine learning . . . . .	44
4.3	Behavioural cloning . . . . .	46
4.4	From traces of behaviour to learner models . . . . .	49
4.4.1	Traces of behaviour . . . . .	50
4.4.2	Pre-processing . . . . .	51
4.4.3	Machine learning program . . . . .	54
4.4.4	Post-processing . . . . .	55
4.5	The nature of the models . . . . .	56
<b>5</b>	<b>Evaluation of the learner models</b>	<b>59</b>
5.1	Evaluation of predictive models . . . . .	59
5.1.1	On-line vs off-line prediction . . . . .	60
5.1.2	Predicting crucial vs predicting non-crucial actions . . . . .	60
5.1.3	Evaluation of contents vs evaluation of performance . . . . .	61
5.2	Evaluation of models of artificial controllers . . . . .	61
5.2.1	Off-line prediction . . . . .	65
5.2.2	On-line prediction . . . . .	66

5.2.3	Contents of the models . . . . .	66
5.2.4	Discussion . . . . .	72
5.3	Evaluation of models of novice human controllers . . . . .	72
5.3.1	Off-line prediction . . . . .	74
5.3.2	On-line prediction . . . . .	74
5.3.3	Contents of the models . . . . .	74
5.3.4	Discussion . . . . .	77
5.4	Conclusions . . . . .	78
<b>6</b>	<b>Presentation of learner models</b>	<b>79</b>
6.1	Understandable learner models . . . . .	79
6.1.1	Presenting the models in the right modality . . . . .	80
6.1.2	Increasing the modularity of the models . . . . .	84
6.1.3	Providing interactivity . . . . .	86
6.2	Evaluation of the interface I . . . . .	89
6.2.1	Procedure . . . . .	89
6.2.2	Results . . . . .	91
6.2.3	Discussion . . . . .	92
6.3	Evaluation of the interface II . . . . .	95
6.3.1	Procedure . . . . .	95
6.3.2	Results . . . . .	97
6.3.3	Discussion . . . . .	101
6.4	Obstacles to model comprehensibility . . . . .	103
<b>7</b>	<b>Testing the effects of participative learner modelling</b>	<b>105</b>
7.1	The effects contextualised . . . . .	106
7.2	Design and implementation of the experiments . . . . .	107
7.2.1	Baseline setting . . . . .	107
7.2.2	Condition . . . . .	109
7.2.3	Testing . . . . .	109
7.2.4	Implementation . . . . .	111
7.3	Expected outcome . . . . .	111
7.4	Baseline performance . . . . .	112
7.5	Posterior performance . . . . .	113
7.6	Performance on the transfer task . . . . .	114



7.7	Articulation of the playing strategy . . . . .	117
7.7.1	The questions . . . . .	117
7.7.2	Translation of the answers . . . . .	118
7.7.3	Evaluation . . . . .	122
7.8	Ranking of state variables . . . . .	125
7.9	Ranking of states of the pole and cart . . . . .	127
7.10	Discussion . . . . .	130
<b>8</b>	<b>Discussion and conclusions</b>	<b>133</b>
8.1	The effects of participative learner modelling . . . . .	135
8.2	A global view of the trail . . . . .	139
8.3	Conclusions . . . . .	141
8.3.1	A new kind of domain . . . . .	142
8.3.2	Concrete hypotheses and empirical results . . . . .	143
	<b>Bibliography</b>	<b>145</b>
<b>A</b>	<b>Dynamics and control of a pole on a cart</b>	<b>161</b>
A.1	Mathematical modelling . . . . .	161
A.2	Control . . . . .	164
<b>B</b>	<b>Material of Study 3</b>	<b>165</b>
<b>C</b>	<b>Material of Experiment 1</b>	<b>175</b>
<b>D</b>	<b>Learner models presented in Experiment 1</b>	<b>209</b>
D.1	Example of inspecting and editing of a learner model . . . . .	223
D.1.1	Inspecting the learner model . . . . .	223
D.1.2	Editing the learner model . . . . .	223
<b>E</b>	<b>Published papers</b>	<b>225</b>

# List of Tables

3.1	A standard set of parameters for the problem of balancing a pole on a cart . . .	34
5.1	The set of rules used by the artificial controllers to control the pole and cart . .	62
5.2	Summary of the performance of the artificial controllers . . . . .	64
5.3	Summary information about the contents of the models of the artificial controllers and their degree of similarity to the original set of rules . . . . .	69
5.4	Summary of one-way analysis of variance of similarity by action delay . . . . .	70
5.5	Model with the best match to the original set of rules . . . . .	71
5.6	A model with average similarity to the original set of rules . . . . .	71
5.7	The model with the worst match to the programmed set of rules . . . . .	71
6.1	Outcome of evaluating the answers to the main questionnaire of Study 3 . . . .	93
6.2	Summary of how much the <b>Expert</b> group used the facilities to run and verbalize rules . . . . .	97
6.3	Answers of the <b>Expert</b> group to questions 3, 4 and 5 of Experiment 1 . . . . .	98
6.4	Summary of how much the <b>Model</b> group used the facilities to justify, run and verbalized rules . . . . .	98
6.5	Summary of learner models and responses of the <b>Model</b> group . . . . .	99
7.1	Summary of results of Kruskal-Wallis ANOVA of the baseline performance of the participants in Experiments 1a and 1b . . . . .	114
7.2	Summary of results of Kruskal-Wallis ANOVA of the posterior performance of the participants in Experiments 1a and 1b . . . . .	116
7.3	Ranks of the state variables, as reported by the participants in Experiment 1a . .	126
7.4	Ranking of the state variables estimated from the number of occurrences of each variable in the learner models of each participant in Experiment 1a . . . .	127
7.5	States of the pole and cart that were ranked by the participants in Experiment 1b	129

7.6 Summary of the results of Experiments 1a and 1b . . . . . 132

# List of Figures

1.1	The learner modelling process . . . . .	12
2.1	Overall view of the human cognitive architecture . . . . .	22
2.2	Example of a component of the learner model in MR COLLINS . . . . .	27
3.1	The pole and cart device . . . . .	32
3.2	Graphical user interface to the simulator of the pole and cart device . . . . .	35
3.3	Histogram that illustrates the frequency distribution of time lag between consecutive user actions . . . . .	38
3.4	Histogram that illustrates the frequency distribution of time lag between consecutive executions of the same action . . . . .	39
3.5	Histogram that illustrates the frequency distribution of time lag between consecutive executions of different actions . . . . .	40
4.1	The method of behavioural cloning . . . . .	47
4.2	A framework for constructing learner models from traces of learner behaviour .	50
4.3	Example of a set of rules induced by RIPPER for controlling the pole and cart .	56
4.4	The final learner model . . . . .	57
5.1	Histogram that illustrates the skewed distribution used to calculate reaction delays for the artificial controllers . . . . .	63
5.2	Off-line predicting power of the models of the artificial controllers and level of noise in testing data . . . . .	65
5.3	On-line predicting power of the models of the artificial controllers . . . . .	67
5.4	Off-line predicting power of the models of all participants in the studies . . . .	75
5.5	On-line predicting power of the models of all participants in the studies, compared to the predicting power of guessing in favour of the most frequent action.	76

5.6	Histogram of the difference per model between the number of rules for <i>push-left</i> and <i>push-right</i> actions . . . . .	77
6.1	Graphical presentation of a learner model . . . . .	82
6.2	Final version of the learner model and its presentation . . . . .	85
6.3	Justification of a rule . . . . .	87
6.4	Verbalization of a rule . . . . .	88
6.5	Palette of operations for exploring and modifying the learner model . . . . .	88
6.6	A fictitious set of rules for controlling the pole and cart . . . . .	90
6.7	Estimated difficulty of the questionnaire used in Study 3 . . . . .	92
6.8	Level of understanding per participant in Study 3 . . . . .	94
6.9	Set of rules presented to the <b>Expert</b> group in Experiment 1 . . . . .	96
7.1	Overall structure of Experiments 1a and 1b . . . . .	108
7.2	Control-panel interface to the simulator of the pole and cart . . . . .	110
7.3	Baseline performance of the participants in Experiments 1a and 1b . . . . .	113
7.4	Posterior performance of the participants in Experiments 1a and 1b . . . . .	115
7.5	Percentage of improvement between baseline and posterior performance among the participants in Experiments 1a and 1b . . . . .	116
7.6	Performance of the transfer task by the participants in Experiment 1b . . . . .	117
7.7	Overall predicting power of the descriptions provided by the participants in Experiment 1a . . . . .	123
7.8	Predicting power of the descriptions provided by the participants in Experiment 1a restricted to <i>push-left</i> and <i>push-right</i> actions . . . . .	124
7.9	Gain in overall predicting power by the descriptions provided by the participants in Experiment 1a . . . . .	124
7.10	Gain in predicting power by the descriptions provided by the participants in Experiment 1a, restricted to <i>push-left</i> and <i>push-right</i> actions, when compared to guessing in favour of the most frequent action. . . . .	125
7.11	Correlation between the estimated and reported importance of the states variables per participant per condition . . . . .	128
7.12	Rankings given by the participants in Experiment 1b to each one of the ten states of the pole and cart they had to judge . . . . .	130
7.13	Correlation between the states' degree of difficulty, estimated and judged by the participants in Experiment 1b . . . . .	131

A.1	Free-body diagram of the pole and cart device . . . . .	162
D.1	Learner model presented to Participant 1 in Experiment 1 . . . . .	209
D.2	Learner model presented to Participant 4 in Experiment 1 . . . . .	210
D.3	Learner model presented to Participant 6 in Experiment 1 . . . . .	211
D.4	The learner model of Participant 6 after being edited . . . . .	212
D.5	Learner model presented to Participant 10 in Experiment 1 . . . . .	213
D.6	Learner model presented to Participant 15 in Experiment 1 . . . . .	214
D.7	The learner model of Participant 15 after being edited . . . . .	215
D.8	Learner model presented to Participant 19 in Experiment 1 . . . . .	216
D.9	The learner model of Participant 19 after being edited . . . . .	217
D.10	Learner model presented to Participant 20 in Experiment 1 . . . . .	218
D.11	Learner model presented to Participant 23 in Experiment 1 . . . . .	219
D.12	The learner model of Participant 23 after being edited . . . . .	220
D.13	Learner model presented to Participant 27 in Experiment 1 . . . . .	221
D.14	Learner model presented to Participant 29 in Experiment 1 . . . . .	222

# Introduction and overview

Scholarship is the process by which butterflies are transmuted into caterpillars.

Jerry A. Fodor (1983). *The Modularity of Mind*.

Learner modelling is the process by which an intelligent tutoring system and learning environment acquires information about its user (a learner), transforms it and incorporates the result into an internal representation of the learner known as the learner model. Traditionally, learner modelling has been a process running behind the scenes, with the learner kept as unaware of it as possible. This dissertation is concerned with a different style of learner modelling: one in which the learner plays an active role and he<sup>1</sup> is aware of it.

The idea of this more *participative* way of modelling the learner has been around for some time, being applied in tutoring systems that deal with subject matters such as reading and writing in native or a second language, computer-based text editing, basic arithmetic skills, scientific terminology and engineering concepts and procedures. Among the main issues that have been investigated are how to involve the learner in the modelling process, how to make overt the content of the learner model, how to incorporate information supplied by the learner into the model, whether this information helps to get a faithful learner model, and ways of using the model as a tool for communication and supporting learning.

It has been a common belief that participative learner modelling in general, and encouraging the learner to inspect the learner model in particular, prompts the learner to reflect on and become more aware of his knowledge. The reason put forward to justify this belief is convincing; the core of it was nicely expressed by John Self in his keynote speech at the Ninth International Conference on Artificial Intelligence in Education:

What makes an inspectable learner model different, of course, is that the person doing the inspecting is also the object of the model . . . An inspectable learner model

---

<sup>1</sup>The convention followed in this dissertation to diminish sex bias in language is to switch from 'he' to 'she' and back from chapter to chapter.

offers the learner a distorted mirror: the learner (if he cares at all) is challenged to consider whether the distortions are due to the system's or the learner's mistaken view of the real situation. (Self, 1999b)

That the learner will reflect on and become more aware of his knowledge follows from the assumption that it is hard to look at one's image in a mirror and not to think about oneself—especially if the mirror prompts you to look carefully. Empirical support for this belief has been accumulating slowly, though, most of it being of an anecdotal character.

The research discussed in this thesis explores the issue of how to interpret learners' reflection and increased knowledge awareness in terms of changes in learner behaviour that can be tested in practice. The opportunity was taken to explore also the generality of participative learner modelling by applying it to a new kind of domain: the acquisition of a sensorimotor skill. In particular, the skill of controlling a pole, or inverted pendulum, attached to the top of a cart ('the pole and cart' for short) is used as a representative instance of sensorimotor skill. The main contributions of the research are:

- i) an extension of the sort of domains participative learner modelling has been applied to, that speaks for the generality of this style of learner modelling;
- ii) a set of concrete hypotheses about the effects on learners of participating actively in the modelling process;
- iii) initial empirical results of testing these hypotheses in the domain of acquiring a sensorimotor skill, and
- iv) a proposal for a methodology, of which this research is an instance, for carrying out further investigations on these issues.

Extending the scope of participative learner modelling to the acquisition of sensorimotor skills is important, not only because it exhibits the generality of the approach, but also because these sort of skills are going to be an important field of application for intelligent learning environments in the near future. New computational technologies, especially new input/output devices, will allow the gathering and intelligent processing of the massive amounts of the fine-grained data required to monitor, guide and convey perceptual and motor behaviour. A sample of this future is already present among us in the form of video-games<sup>2</sup> and the growing interest in applying artificial intelligence techniques to make them more attractive.

---

<sup>2</sup>Even if one does not agree, as I do not, with the overwhelming violence "the best" of these parade.



The paragraph above may give the false impression that selecting a sensorimotor skill as test-bed domain was dictated by an interest in selling the idea of participative learner modelling in these domains. That is not the case in fact. The reason for choosing control of the pole and cart as the subject matter was that evidence from research in this and similar tasks indicates the necessary skills can be learnt and optimised without their possessor being able to articulate them reliably. For the purposes of this research, that means a learner can be considered a *tabula rasa* at the start of participative learner modelling, and so any effects of the latter should manifest themselves as clear marks on the otherwise pristine surface.

Participative learner modelling covers a wide range of possible ways the learner may assume an active role in the learner modelling process. For example, he can collaborate, co-operate, negotiate or take full control of what is stored in the learner model; he can supply information explicitly, filter information gathered by other means and have a say on how the information is used to support his learning. Encouraging the learner to inspect and edit his learner model is, nevertheless, a prototypical instance of participative learner modelling, and the one this dissertation focuses on when there is a need to be more specific. In particular, it is this form of learner modelling that is implemented by the collection of programs that apply participative learner modelling to the domain of controlling the pole and cart—programs that are hereafter referred to by the umbrella name of PACMOD.

In this research a series of studies was conducted using control of the pole and cart as test-bed domain. The overall outcome of these studies suggests that participative learner modelling makes learners become better at articulating their knowledge; a fact that can be interpreted as evidence to support the claim that this form of learner modelling actually promotes learners' reflection on their knowledge and increases knowledge awareness. On the other hand, no evidence was found that participative learner modelling leads to other enhancements in the access a learner has to his knowledge; improved access that would allow him to transfer his skill on the original task to other closely related tasks or to generally employ his knowledge in more flexible ways (e.g. to perform other, more different tasks in the same domain). The results are, however, far from conclusive, leaving ample space to explore in future investigations in the area.

Opening the process of learner modelling to learner participation sounds like a good thing to do, especially since it coincides (both conceptually and temporally) with a trend to *openness* in research in Artificial Intelligence in Education. This trend is apparent in the current emphasis on learning, exploration and collaboration, as opposed to teaching, curriculum and system adaptation to the individual learner. After all, this is the era of the Internet, the epitome of

openness. It is very important, in the face of this state of excitement, to examine the underlying assumptions and weigh the strengths and limitations of a participative approach to learner modelling. Otherwise, we may end up selling something we cannot deliver.

## Overview of the dissertation

After this introduction and overview, the dissertation continues in Chapter 1 with a closer look at participative learner modelling. It presents participative learner modelling as a way to lessen the difficulty of constructing learner models which also promises additional benefits (e.g. the learner model as an educational tool). The chapter continues with a review of previous research, before addressing the issue of participative learner modelling as a promoter of reflection and knowledge awareness.

Chapter 2 starts with summaries of the views of the human cognitive architecture, skill acquisition and meta-cognition that serve as the theoretical background for a subsequent analysis of reflection and knowledge awareness in the context of participative learner modelling. The analysis concludes with the effects that reflection and awareness should have on learner's knowledge; namely, the construction and reinforcement of conceptual knowledge that is accessible to meta-cognition and can be used in different ways to cope with a variety of new situations and tasks. Then a hypothesis is stated for changes in learner behaviour that should manifest themselves as a consequence of the learner being actively involved in learner modelling (and hence reflecting and becoming more aware of his knowledge): increased ability to report their knowledge and use it to perform new tasks, possibly accompanied by a decrease in their performance at well-practised tasks (as conceptual knowledge may interfere with learners' procedural knowledge of these tasks).

Modelling learners demands a domain to be learnt; it cannot be done in vacuum. Toy domains or micro worlds have properties that make them excellent domains for experimentation: generally well defined, with explicit and clear descriptions of what is to be learnt (usually less than in "real life" domains), the goals to be pursued and the range of possible behaviours. One of this toy domains is described in Chapter 3, that centres on the task of controlling the pole and cart. The chapter includes a description of the domain and the operation of the computer-based simulation of the pole and cart used throughout the research. A justification is given in this chapter of why control of the pole and cart is appropriate as a test-bed domain for exploring participative learner modelling and its effects. The chapter concludes by presenting relevant information about human performance on this and other sensorimotor tasks.

Chapter 4 contains a description of the method for learner modelling developed as part of the research. The method is based on a technique known as *behavioural cloning* for acquiring expert knowledge from traces of expert performance; it builds models that are sets of production rules induced from traces of learner behaviour by means of machine learning. The chapter ends with a discussion about the properties of the learner modelling method and the models that it produces.

The description of the design and implementation of an interface to the learner models is postponed until Chapter 6. The central issues discussed there are additional refinements of the contents of the learner models necessary for making them more comprehensible to learners, as well as the relationship between the modalities used for presenting the main tasks in the domain and the learner model. The necessity of providing facilities for justifying and illustrating the contents of the models, as well as facilities for learners to modify the models are also discussed in this chapter.

A set of empirical studies was necessary to test the reliability of the learner modelling technique, the quality of the learner models and the learners' understanding of them. In other words, it was necessary to justify that learners would be confronted with *models* of themselves that they would be able to understand without too much effort. The studies concerning the reliability of the modelling technique and the quality of its products are reported in Chapter 5; the studies that evaluated the clarity of the presentation of the models are reported at the end of Chapter 6.

Two experiments were designed and conducted for testing out the hypothesis stated in Chapter 2 for changes in learner behaviour consequent to participative learner modelling. They are reported in full detail in Chapter 7. Each experiment involved three stages: controlling the pole and cart, performing a condition task and then performing a set of test tasks. The condition task was either controlling the pole and cart for a longer period of time (i.e. more practice), inspecting and editing a suggested "good" strategy for doing the control (an analogue of teaching that is very similar to participative learner modelling) or inspecting and editing the learner model. The test tasks were all in the same domain of the pole and cart but included reporting the control strategy used, ranking properties of the pole and cart by its relevance to control, ranking states of the pole by the difficulty they pose to control, and controlling the pole and cart using a different interface to the simulator described in Chapter 3.

Chapter 8 starts with a brief account of the investigation as a whole, and continues with a detailed discussion of the decisions taken along it, the advantages and drawbacks of these decisions and the alternatives discarded that deserve exploration. The characteristics of the

proposed methodology for exploring participative learner modelling are discussed afterwards, before the chapter ends with a statement of the overall conclusions of the research and suggestions for further work.

The dissertation includes addenda with more details on the physics and control of the pole and cart (Appendix A), the materials used in the studies and experiments and data that complements their reports in previous chapters (Appendices B and C), copies of published papers (in Appendix E) and bibliography.

# Chapter 1

## Participative learner modelling

Destructive criticism is one of life's great pleasures, and a seriously undervalued one.

Richard Palmer (1993). *Write in Style: A Guide to Good Style*.

This chapter begins with an explanation of what is meant by 'participative learner modelling' and continues with a review of previous research in the area. The common view that participative learner modelling promotes learner's reflection on and awareness of her knowledge is analysed, in order to exhibit a gap in its basic assumptions that merits further investigation.

### 1.1 Learner modelling

Intelligent tutoring systems and learning environments have been designed, to a large extent, to be able to keep an explicit representation of their knowledge. In particular, they have often been capable of storing, maintaining and exploiting a structured representation of their user (a learner); a subset of their knowledge called their 'learner model' (or 'student model'). The problem of learner modelling can be stated—in a very ambitious way—as the problem of representing, explicitly and faithfully, all aspects of the learner that concern her learning (knowledge and misconceptions, cognitive abilities and deficiencies, preferred learning styles, degree of self-confidence, mood and other emotional traits, etc.). The representation should be tailored to each individual learner and updated constantly to reflect changes in her knowledge and state of mind, capabilities and personality. Finally, the system should obtain all relevant information without disturbing, but rather enhancing, its interaction with the learner<sup>1</sup>.

---

<sup>1</sup>From now on, the term 'Intelligent Learning Environment' (ILE) will be preferred to 'Intelligent Tutoring System' (ITS), and the term 'learner model' will be preferred to 'student model'. The main reason for these preferences

The paragraph above poses an extremely difficult problem for a computer system, acknowledged in the famous paper ‘Bypassing the intractable problem of student modelling’ by Self (1988a). He insists, however, that learner modelling is a key feature of any truly *intelligent* learning environment, and suggests that the right approach to coping with the sheer difficulty of the problem is not to cast learner modelling aside as intractable but to reconsider its philosophical roots, theoretical basis, educational strengths and practical implementation. He proposes a number of ways of getting around the intractability of the problem, which can be summarised in four directives:

- i) model the learner only to the level of detail required to support and improve teaching and learning;
- ii) provide a user interface rich in interactive facilities in order to reveal the cognitive state of the learner;
- iii) regard the learner model as representing the learner as she is, without judgements of value based on comparisons with models of expertise singled out more or less arbitrarily; and
- iv) allow the learner to participate actively in the modelling process.

In contrast to the traditional approach of regarding the learner as an object to be diagnosed as surreptitiously as possible, the last directive above encourages the *active* and *explicit* involvement of the learner in the modelling process: a style of learner modelling that can be referred to as ‘participative’. It does not matter whether the learner’s goals are similar, different, or even contrary to the goals of the system and other participants in the modelling process; nor does it matter how the task is distributed among the participants, nor their level and timing of interaction. It is in this sense that to participate is less restrictive than to collaborate, to cooperate and to negotiate, the latter being in fact specialisations of the former (Baker, 1994; Teasley & Roschelle, 1993).

Asking the learner for help in order to simplify the learner modelling problem does not make it less interesting. On the contrary, many issues remain to be investigated, and Self (*ibid.*) touches on some of them. For example, there are the issues of how to design user interfaces that gather detailed information about the learner’s cognitive processes in a non-intrusive way (Salvucci, 1999a; Twidale, 1992); how to represent the evolving set of the learner’s beliefs,

---

is generality: an ITS can be thought of as an ILE with a pervasive tutoring component, and a student is (hopefully) a learner; however, not all ILEs aim to *teach*, nor all learners are students. It is also the case that ‘intelligent learning environment’ and ‘learner model’ are options somewhat more “progressive”—although ‘intelligent tutoring system’ and ‘student model’ are better known terms.

specifically beliefs about their own state of knowledge (Self, 1994a,b); how to give psychological credibility to learner models (Anderson et al., 1995); how to design systems capable of maintaining learner models in collaborative interactions with learners (Bull, 1997a; Dimitrova et al., 1999a,b), and how to convey a convincing image of the system as a collaborator (Dillenbourg & Self, 1992; du Boulay et al., 1999; Hietala & Niemirepo, 1997).

Participative learner modelling provides the learner with additional learning experiences and equips the system with another channel to interact with the learner. This is important because it increases the benefits of designing, implementing and running a (costly) learner-modelling strategy. From this viewpoint, a learner model acquires new purposes, beyond the conventional ones of adapting the system to the particular needs of the learner, predicting her behaviour and assessing her knowledge—Self (1988b) provides a description of these and other established uses of a learner model.

Self (1988a) praises open models for their potential to support, provoke, and even challenge learners to reflect on their own understanding. He believes open models could help learners to ‘develop a more favourable self-image and a better view of how knowledge is acquired’ (see also Dillenbourg, 1992). Furthermore, he suggests that designing learner models to be inspectable by learners could be a ‘salutary principle’ that ‘might benefit ITS design by reducing the temptation to include crude, ad-hoc classifications’ in the models.

## 1.2 External influences

Learner modelling does not occur as an isolated phenomenon. On the contrary, it responds to a number of external forces which exert their influence in the evolving conceptions of what a learner model is for, how it can be built, what its content should be and in what sense it represents the learner.

The shift from covert to participative learner modelling parallels the change of focus from tutoring systems to learning environments. Whereas the existence of learner models in intelligent tutoring systems reflects the systems’ attention to each learner’s individuality (Ohlsson, 1986; Self, 1999a; Wenger, 1987) most of them embody direct instruction of a predefined and well structured curriculum. Intelligent learning environments, on the other hand, generally incorporate a learner-centred philosophy that conceives learning to be constructing knowledge from personal learning experiences gathered through independent exploration of the subject matter (Hannafin & Land, 1997). Endowing learners to take control of their learning experiences makes it harder to maintain a learner model almost exclusively from information gath-

ered in a furtive manner—for example, the system cannot choose the next exercise in order to discern between competing explanations of learner behaviour.

In a broader interpretation of learner control, it can be argued that learners should be granted access to their learner models and the rights to discuss their content and ask for suitable modifications (Kay, 1997). This is so on the same basis as legislation in some countries demands that any personal data about individuals that is kept in a computer be registered, and gives individuals the right to access their data and to have it corrected or deleted—e.g. the Data Protection Act in the UK (Data Protection Registrar, 1999).

Learners have to be able to deal with their new responsibility as directors of their own learning process; otherwise they will get lost in the middle of rich and flexible learning environments, not knowing which path to explore nor which question to ask, unable to learn anything from either success or failure (e.g. Aleven & Koedinger, 2000)—a version in a micro-world of the problem we face, overloaded with information, of finding our way in the macro-world of modern societies. An approach to this problem, that does not fall back on taking control away from the learners, is represented by the emphasis that some modern educational theories place on meta-cognition (Weinert & Kluwe, 1987). In the same way that cognition refers to the capability of acquiring knowledge about the world, processing this knowledge and using it for problem-solving, meta-cognition is about acquiring knowledge about ourselves (and other cognitive creatures), processing this “meta-knowledge”, and using it for directing, monitoring and evaluating our problem-solving activities. In essence, the idea is that learners should be able to exert control over their knowledge acquisition and problem-solving, and hence increase the efficiency and quality of their learning, by improving their meta-cognitive abilities (Joyce et al., 1997). Participative learner modelling fits better in this view of education than covert learner modelling because it gives learners an opportunity to rehearse and improve their self-knowledge.

### 1.3 Objectives

Since the publication of Self’s paper (1988a), other researchers have also been driven by the belief that participative learner modelling encourages learners to reflect on and become more aware of their own knowledge, learning and problem-solving. For example, the goal of increasing learners’ self-knowledge, particularly through making their models *scrutable*, was strongly advocated by Kay (1997) in her keynote address at the International Conference on Computers in Education. Promoting reflection in learners is one of the reasons given by Paiva et al. (1995)



for ‘externalising’ learner models. Work done by Bull on ‘collaborative learner modelling’ has been greatly motivated by the aim of helping students of a second language to develop language awareness (Bull, 1997a; Bull et al., 1995), and to reflect on their own knowledge, learning styles and learning strategies (Bull, 1997b; Bull, in press; Bull & Ma, submitted). Her related work on jointly constructing writer models seeks to raise writers awareness about their writing strategies (Bull & Shurville, 1999).

Naturally, participative learner modelling has been seen also as a way of constructing better models—an interpretation of Self’s recommendation of directly asking learners for information instead of struggling to guess it. The idea is exploited by Beck et al. (1997) to augment the power of their student models to predict student performance in maths, by asking fifth-grade students their level of confidence in their knowledge. The UM toolkit for constructing user models (Kay, 1994b) includes facilities for incorporating new evidence, input directly and explicitly by users, which is then considered by the system to conclude whether they know something or not. SCRAWL (Bull & Shurville, 1999) fills a writer model by asking writers questions about their writing process; the model is then used to guess the writers’ writing style. A similar approach, of total trust in the learner, is followed by Bull & Ma (submitted) and de Buen et al. (1999).

A further source of motivation for participative learner modelling has been to decrease the cost to benefit ratio of learner modelling by devising further ways of exploiting learner models. Open models have been either used or proposed as tools for communication between students and the system (Bull & Ma, submitted), between students and teachers (Brna et al., 1999; Bull, 1997b; Pain et al., 1996) and among students (Ayala & Yano, 1996; Bull & Broady, 1997).

## 1.4 A rough classification of approaches

Learner modelling can be made more participative by allowing the learner to play a more active role in any of its three main subprocesses: learner diagnosis, model maintenance and model exploitation (Figure 1.1). Learner diagnosis is the process of obtaining information about the learner to be included in the learner model. Model maintenance refers to the process of encoding such information and actually incorporating it into the learner model. Finally, the learner model is used, in one way or another, for supporting learning—more detailed overviews of learner modelling can be found in (VanLehn, 1988) and (Greer & McCalla, 1994).

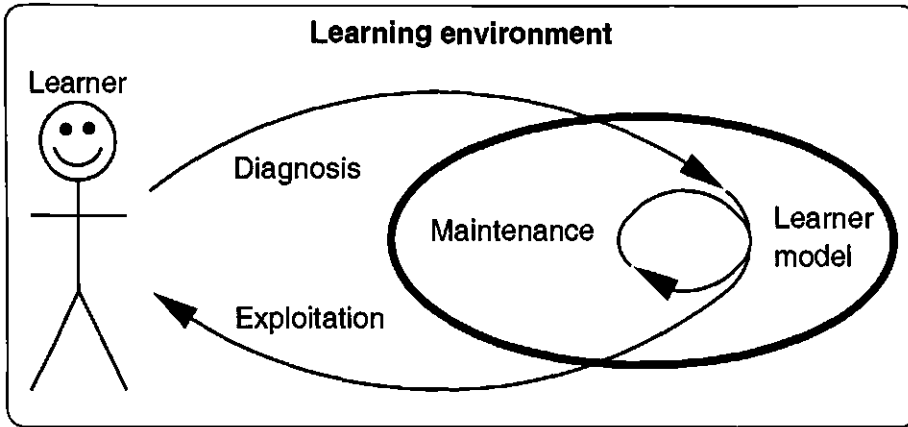


Figure 1.1: The learner modelling process.

#### 1.4.1 Participative learner diagnosis

Covert approaches to learner diagnosis, frequent in such classical work on learner modelling as BUGGY (Burton & Brown, 1982) and the LISP TUTOR (Reiser et al., 1985), attempt to infer information about the learner from unintrusive observations of his behaviour and without him being aware of that happening.

Alternatively, data about learners can be collected by asking learners explicit questions. The classic example in the field of user modelling is GRUNDY (Rich, 1979), which asked its users questions like “I’d like to know what sort of person you think you are,” in order to select a stereotype that was then used for recommending books. In GRACILE (Ayala & Yano, 1996), learners of a second language state explicitly their learning goals and task commitments (i.e. tasks they are willing to carry out for solving a shared problem or assisting other learners). A modified version of the MFD system (Beck et al., 1997) asked children how confident they were of having basic mathematical skills with whole numbers and fractions. Undergraduate students using SCRAWL (Bull & Shurville, 1999) are requested *ab initio* to answer a few questions on their writing habits, from which responses SCRAWL infers their stereotypical writing style. A similar approach is followed in LS→LS (Bull & Ma, submitted), which asks questions about learning styles and language learning strategies.

There is a subtle but important difference between the previous examples of learners explicitly providing information about themselves—the kind of person they are, their learning goals and commitments, level of confidence on their knowledge, writing habits, learning style and favourite learning strategies—and learners providing information to the LISP TUTOR (Reiser et al., 1985), the GEOMETRY TUTOR (Anderson et al., 1985) and EPIC (Twidale, 1992), for

example. In the former cases, learners can easily become aware they are involved in the construction of a learner model—what would the system want this information for, otherwise?—whereas in the latter cases, no matter how detailed the information supplied by the learners is, the chance is faint for them to feel engaged in any other task besides trying to solve their problems.

#### 1.4.2 Participative model maintenance

The fact that learners can participate in the diagnosis subprocess of learner modelling does not guarantee that the content of the learner model will be accessible to them, nor vice versa. In GRUNDY and MFD, for example, the model is kept away from the user; PACMOD <sup>2</sup>, on the contrary, diagnoses the learner behind the scenes but opens the resulting model to inspection and modification (Chapters 3, 4 and 6).

There are several ways in which a learner can interact with her learner model, and with the system that maintains it. The simplest option is for the system to externalise the model in a suitable way (Paiva et al., 1995) and for the learner to browse, inspect or scrutinise it. Alternatively, the learner may be granted the right to modify the model at will (Bull & Shurville, 1999; de Buen et al., 1999). These two approaches are extreme cases of participative model maintenance in a couple of respects:

1. The control over the content of the learner model falls completely either on the learner or on the system.
2. The interaction between the learner and the system, for modelling purposes, is minimal: the system simply “opens” the model and supplies the facilities for the learner to act at will, in her more or less limited scope of action. No further questions are asked, either because there is no need for them (the learner can do nothing else but observe the model) or because the system trusts the learner blindly.

These approaches correspond to a notion of openness that Self (1999b) regards as ‘too passive, too bland, and too neutral’. A number of richer variations have been thought of, however. The UM toolkit (Kay, 1994b), for example, allows the learner to register her beliefs in her knowledge as part of the learner model<sup>3</sup>, and the system uses this information as evidence,

<sup>2</sup>This is the umbrella name given to a collection of programs developed as part of this research that implements participative learner modelling. It is described in more detail in Chapters 3, 4 and 6.

<sup>3</sup>Although UM is a toolkit for user modelling, its best known application is to model users acquaintance with the SAM editor; a task that can be easily seen as of learner modelling.

either in favour of or against its own beliefs; the importance given to it depends on the use of one ‘resolver’ (conflict resolution procedure) or another. A different approach is illustrated by MR COLLINS (Bull, 1997a; Bull & Pain, 1995), which keeps two separate sets of beliefs, or ‘confidence measures’, about the learner’s mastery of a second language; one set is controlled by the system and the other set is controlled by the learner. Although conflicting beliefs can co-exist in the learner model, MR COLLINS allows the learner to challenge its beliefs. MR COLLINS responds to these challenges either supplying pertinent justifications, negotiating a confidence measure of compromise or asking the learner to prove their knowledge of the language. More recently, Dimitrova et al. (1999a) have been exploring ways to support more flexible learner-system interactions during model maintenance. Their approach, based on models of human-human interaction, considers a richer set of actions than previous research did (inform, inquire, challenge, withdraw, justify, agree, suggest and deny).

### 1.4.3 Participative model exploitation

An important motivation for participative learner modelling has been to exploit learner modelling in novel ways (Section 1.3). Among these new ways of taking advantage of learner modelling have been the use of models as tools for communication between tutoring systems and learners (Ayala & Yano, 1996; Bull, 1997b; Bull & Broady, 1997; Bull & Ma, submitted), to support collaborative assessment between learners and tutors (Brna et al., 1999) and as a tool for learners to communicate with the software agents that represent them in learning environments conceived as ‘hybrid societies’ (Vassileva et al., 1999a).

Holden & Kay (1999) have recently proposed extending the *scrutable* framework of the UM toolkit to the design of *scrutable* teaching modules—an illustrative example of how the participative approach can be extended beyond learner modelling to other components of learning environments.

## 1.5 Research issues

Learner modelling can be a very complex process and produce quite elaborate representations of learners. Consequently, it opens several opportunities for the active and explicit involvement of learners. More importantly, perhaps, it raises a number of interesting questions such as which of these opportunities are worth taking; to what extent learners can, should or even must be allowed to participate in the modelling process; and the implications of opening a particular aspect of learning modelling for the process as a whole, and for the design of other components

of learning environments.

Taking part in the modelling process requires time and attention from learners. Every time they are asked a question about themselves, like how well they think they understand dividing whole numbers, or browse the learner model, they stop performing other activities they may regard as more important to their learning of the subject matter. They can become quite frustrated (Beck et al., 1997). Obvious as it may sound, appropriate timing of interruptions, careful selection of questions, and the provision of effective and efficient user interfaces<sup>4</sup> are critical in order to diminish learner distraction and irritation.

Learner models can contain information that may be very difficult to present in a clear form, may overload or confuse learners, or may even be too sensitive for learners to know about (e.g. how much the system trusts the learner's level of confidence in her knowledge; Beck et al., 1997). Non-symbolic and mixed learner models (Jameson, 1996), for example, may pose serious difficulties for visualisation (cf. Zapata-Rivera & Greer, 2000); the same is true of highly complex symbolic representations. Presenting in full detail the evidence that supports inferences based on machine-learning techniques (Chapter 4; Webb & Kuzmycz, 1996) may be unnecessary or undesirable. Stereotypes can be quite useful and effective (Kay, 1994a), but may be interpreted in a pejorative sense by some learners. From another point of view, as suggested by Self (1988a) and (Kay, 1997), learner participation in the modelling process can be seen as a beneficial principle of design, even if not fulfilled in practice. Moreover, as was mentioned in Section 1.2, current and future regulations may demand each bit of personal information stored in the learner model be accessible to learner's scrutiny. The same regulations may also give learners the right to put limits on the availability of that same information to other parties.

A caveat against giving learners voice in the learner modelling process comes from the well-known attack on introspection in the early part of the past century<sup>5</sup>, as described in (Ericsson & Simon, 1984), and from more recent studies on the quality and quantity of learners' level of self-knowledge (Barnard & Sandberg, 1996): not only can learners' self-knowledge be very little and inconsistent with their performance (e.g. they may think they know how to do something, but being unable to do it and to articulate their knowledge) but also they may not be interested in improving it, or may find the task too hard. Furthermore, even if learners do have a good deal of accurate self-knowledge, they may express it using a different set of

---

<sup>4</sup>Chapter 6 elaborates upon this point, using the interface designed for this research as a reference.

<sup>5</sup>I acknowledge there is no agreement on whether this year of 2000 marks the end of the 20th century or the beginning of the 21st!

conventions—‘How can we be sure that the introspecting observer uses language in the same way as the interpreting experimenter?’ (Ericsson & Simon’s phrasing of Watson’s (1913) attack on analytic classical introspection, *ibid.*, p. 58; more on this in Sections 7.7 and 7.10). Other researchers have suggested that blindly empowering learners to take control of their learning may be detrimental because they may lack the necessary (meta) knowledge to make proper use of this power (Aleven & Koedinger, 2000). Nonetheless, no research has been done which shows that participative learner modelling can hamper learning or is less beneficial than covert modelling. This fact can be interpreted in a number of ways, though: as a tacit approval of the tenets and results of this style of learner modelling; as a sign of the strength of the current trend to openness in intelligent learning environments, mentioned in the Introduction; or as a symptom of the little attention that participative learner modelling has received up to now.

A viable way of getting around the lack of learners’ self-knowledge follows from Ericsson & Simon’s insight that useful and well-supported information can be obtained from verbal reports without needing to trust the speaker. For example, the learner may be mistaken when she affirms that she knows how to do something (e.g. how to divide whole numbers), but her affirmation implies that she is at least aware of the *existence* of such a piece of knowledge (i.e. a procedure for dividing whole numbers). In a broad sense, this is the approach followed by Beck et al. (1997), who multiply the learner’s level of confidence in her knowledge, as reported by the learner, by a personalised factor of trust that is completely controlled by the system. Another possibility is to use further sources of information (e.g. data from an eye-tracking system, as in Gluck et al., 2000) to qualify the learner’s contributions to the modelling enterprise.

The latter approach gets safer at the expense of reducing learners’ power to make decisions on the modelling process, and can be accused of being disguised diagnosis. It may also have a detrimental effect on the motivation of learners to participate in the modelling process. The opposite approach, that of fully trusting learners (see examples in Section 1.4.2), goes more on the lines of giving learners more control and responsibility by presenting the system as a peer collaborator. Under these conditions, however, a learner model cannot be interpreted anymore as a more or less faithful image of the learner, but as an image the learner *believes* is faithful, or even as the image she *wants* to present to the system. Whether this sort of “learner model” can still be a source of information about the learner, and how a learning environment can make use of it, are open questions (cf. Vassileva et al., 1999a). The idea departs radically from the standard aim behind a learner model: that of being a faithful representation of the learner useful for adapting the environment to better support learning. Other uses of the learner models, as the ones outlined in Section 1.4, still seem to assume there is a certain degree of faithfulness in the

models. Whereas it could be the case for a completely deceived tutor to still provide adequate support to their tutees, it is hard to believe that this would happen frequently. Alternatively, a very clever tutor could “meta-reason” about the model (e.g. ‘she wants me to believe that she knows it,’ or ‘she might believe that I believe that she knows it’) and act accordingly<sup>6</sup>. Some variations of this sort of reasoning have been proposed for learner modelling (Dimitrova et al., 2000; Self, 1994b).

Intermediate solutions should distribute trust, control and responsibility among the learner, the system, and any other participants in the modelling enterprise. Mechanisms for reaching agreement and conflict resolution among the parties should exist, from recording all opinions and executing predefined conflict-resolution procedures, to allowing for full-fledged negotiation to take place. The UM toolkit, for example, implements the first strategy, whilst MR COLLINS combines recording of all opinions with support for challenge and negotiation. The more recent work by Dimitrova et al. (1999b) on STyLE-OLM is an attempt to build a learner modelling component able to collaborate with the learner through dialogues similar in structure to human-human dialogues. STyLE-OLM keeps the beliefs of learner and system separate, and uses formal reasoning to infer shared and conflicting beliefs (Dimitrova et al., 2000).

## 1.6 Participative learner modelling as educational activity

Most research on participative learner modelling has been motivated by the anticipation of educational benefits: participative learner modelling does not distract learners from their primary objective of learning because they actually learn through it—they reflect on, and can be more aware of their own knowledge. Nevertheless, little empirical evidence has been gathered up to now to support this belief. Research has focused on testing whether learners are willing to take part in the modelling process (Bull & Pain, 1995; Cook & Kay, 1994), whether the interface to the learner model is appropriate (Paiva et al., 1995; see also Chapter 6), whether learners’ contribution leads to faithful, predictive models (Beck et al., 1997; Bull & Shurville, 1999), and whether learner models can be useful tools to support human-human collaboration and communication (Ayala & Yano, 1996; Bull, 1997b).

---

<sup>6</sup>That is how The Man in Black defeated Vizzini the Sicilian in the film *The Princess Bride* (Reiner, 1987). Dan Egnor applied these ideas to the design of Iocaine Powder, the program that won the First International RoShamBo [rock-paper-scissors] Programming Competition in 1999, and finished third in this year’s Second Competition (Billings, 2000). As a point of reference, the programs ACT-R PLUS and ACT-R LAG2 competed this year and ended in 14th and 31st place, respectively, out of 64 competitors—ACT-R LAG2 is described in (Lebiere & West, 1999). This simple fact hints to the power of meta-cognition.

Some anecdotal evidence has been reported in support of the educational function of participative learner modelling. Cook & Kay (1994), for example, mention unsolicited comments from a participant in one of their experiments, who considered that browsing the learner model had helped her to identify elements at the frontier of her knowledge and used this information to guide her exploration of the domain. In the same vein, Beck et al. (1997) comment on preliminary evidence of an increase in self-confidence among the female participants in their experiments. Recent versions of tutoring systems based on the ACT-R cognitive architecture (Anderson et al., 1995; Koedinger & Anderson, 1997) include a graphical summary of the learner model called the ‘skill-meter’; however, learners tend to pay little attention to it, and when they do so it appears to have mixed impact on their behaviour, like motivating them to put more effort in the task, but also making them reluctant to ask for help when needed (Koedinger, priv. comm.).

In the case of MR COLLINS, the claim that it promotes reflection and language awareness is put forward on the grounds of the overall design of the system and the transparency of the learner model. Learners *must* reflect on their knowledge of a second language (European Portuguese) and be more aware of it because

- MR COLLINS explicitly shows the rules of the foreign language it teaches, the learner’s confidence on her knowledge (as stated by the learner) and its own judgement of the learner’s mastery of the language;
- MR COLLINS encourages the learner to challenge its beliefs and to negotiate a common estimate of her knowledge; and
- learners actually challenge MR COLLINS and try to reach an agreement (Bull, 1997a; Bull et al., 1995; Bull & Pain, 1995).

More recently, Bull & Ma (submitted) report that awareness of language learning strategies, the subject matter of their LS→LS system, increased among the participants in their studies. They also report cases of increased awareness of the learning process. It has to be noted, however, that the final stage of the interaction with LS→LS, in which the system suggests learning strategies not yet used by the learners, resembles more a teaching activity than a modelling one—even if the suggestions are in fact stored in the learner model. In consequence, it is unclear to what extent the increase of awareness among learners can be attributed to their active role in the modelling process. Furthermore, the sample size of the studies was rather small (only four participants; the same participants in both studies).



It sounds obvious that a learning environment that supports participative learner modelling is different to a learning environment that uses covert learner modelling: in the former, the learner can become aware not only of the existence of a learner model and of the special relationship of the model to her—something that could happen in the latter too—but also of its content. In a sense, it is like seeing her image in a mirror; and it is certainly hard to look at one's image in a mirror and not think about oneself. Concluding from this that learners involved in the modelling task will reflect on their beliefs and become more aware of them is, nevertheless, a leap that requires empirical validation, no matter how self-evident it may appear. A similar remark is made by Dillenbourg (1992, p. 196):

The availability of reflection tools does not guarantee that users do indeed reflect on their learning experiences.

Many things can go wrong: the learner may fail to realise what she is involved in; she may take part in the modelling process without being engaged in it; she may not understand the information she has access to, etc.

## 1.7 How to build a bridge to Utopia

The previous section poses a question: does participative learner modelling promote learner's reflection on and awareness of her knowledge? As it is, the question is too general, and can be compared to another question: can we build a bridge to Utopia?<sup>7</sup> This dissertation describes a way of answering the first question that is equivalent to answering the second one by building the bridge.

It has been suggested that participative learner modelling is a good way of promoting learner's reflection and knowledge awareness. In order to explore the possibilities of that being so, it is convenient to reduce the scope of research by giving the learner a well defined role in learner modelling—a decision equivalent to choosing a place to start building the bridge. Encouraging the learner to inspect and modify (if necessary) her learner model has been the most common, and one of the simplest forms of participative learner modelling to date (Section 1.4). Consequently, it is a good starting point for an exploration that attempts to throw some light on a general issue in the area, and so this dissertation recurs to this particular instance of participative learner modelling when there is a need to be more specific. Furthermore, it is necessary

---

<sup>7</sup>Actually, I am not referring here to the imaginary and ideal country in *Utopia* (1516) by Sir Thomas More, but to the slightly less imaginary place, somewhere across the puddle and full of delicacies, in the film *Antz* (Darnell et al., 1998).

to interpret reflection and knowledge awareness in more concrete terms, up to the point where it is possible to design experiments for testing their occurrence—i.e. we need a specific target for the bridge to aim at. The interpretation given to the phrase ‘reflection and knowledge awareness’ is explained in the next chapter.

As much as building a bridge requires some material and a way of putting it together, learner modelling of any sort demands a subject matter to be learnt and a method for constructing the learner models. In addition, if the models are going to be opened to learners’ inspection and modification, it is necessary to implement the means for making the models accessible to the learners. Chapters 3 to 6 describe and explain the decisions taken along the process of building a system called PACMOD that implements inspectable learner models (see Footnote 2 in this chapter), as well as a series of studies conducted for evaluating this system. Afterwards, Chapter 7 contains a description of the final experiments and their results—i.e. whether the learners ended up in Utopia, or not.

## Chapter 2

# Effects of participative learner modelling on the learner

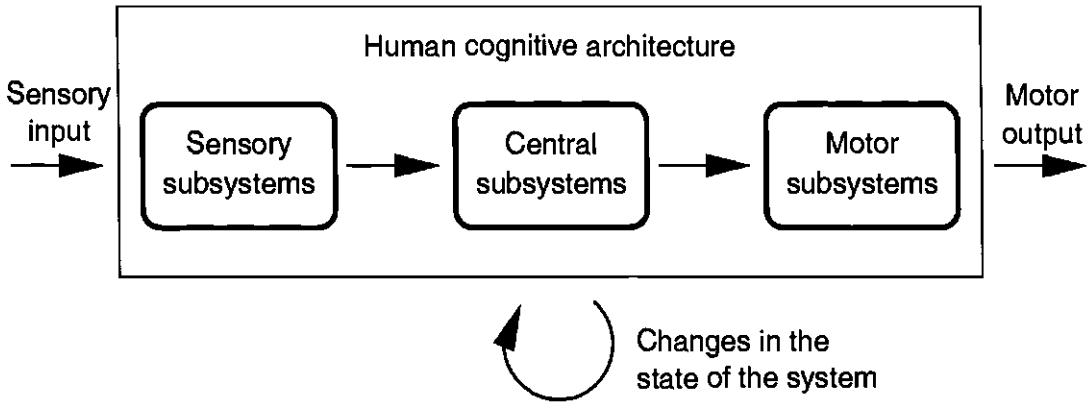
Eventually, I believe, current attempts to understand the mind by analogy with man-made computers that can perform superbly some of the same external tasks as conscious beings will be recognized as a gigantic waste of time.

Tomas Nagel (1986). *The View from Nowhere*.

This chapter elaborates on participative learner modelling as a promoter of reflection and awareness, to produce a set of hypotheses of consequent changes in learner behaviour that can be tested empirically. It starts with quick overviews of the “classical” theory of the human cognitive architecture, skill acquisition and meta-cognition, which provide the language for the subsequent discussion of the effects of participative learner modelling. The first section roughly follows the presentation given in (Stillings et al., 1995, Ch. 1–3).

### 2.1 The human cognitive architecture in a nutshell

The now “classical” view of the set of built-in mechanisms responsible for the mental processes in humans that lead to knowing, learning and understanding things (i.e. to *cognition*) is that they make a system for processing information (Newell, 1990; Stillings et al., 1995). Furthermore, the system is regarded as decomposed into three types of subsystems, or modules: sensory, central and motor (Figure 2.1). Information enters into the system through the senses (vision, hearing, touch, smell, taste, proprioception), it is processed as it flows through the sensory, central and motor subsystems, and finally produces motor actions and changes in the



**Figure 2.1:** Overall view of the human cognitive architecture

state of the system. Sensory modules transform input into appropriately encoded information, to be processed by central modules; central modules store and process information, and deliver appropriately encoded commands to motor subsystems; and motor subsystems transform these commands into muscle contractions and extensions, resulting in movement—see (Byrne & Anderson, 1998; Meyer & Kieras, 1997a) for recent developments of the general framework. The classical view regards sensory and motor modules as informationally encapsulated and cognitively impenetrable, in the sense that their internal mechanisms are seen as mostly independent of, inaccessible to, and unaffected by processing in the central subsystems (Fodor, 1983).

Built-in mechanisms operate in central subsystems to store, reorganise, strengthen, tune, retrieve, interpret and execute information. The information produced by sensory subsystems is stored in *conceptual* (or *declarative*) *memory* using a *propositional representation* and organised as an *associative network*. Propositions in conceptual memory are organised in *chunks* to represent either individual facts (e.g. that an apple is red) or more general concepts like the general notion of what a game is (a *concept schema*) or what it is like to be in a restaurant (an *activity schema*, or *script*). Most of this information is normally in a state of quiescence, active enough only to “survive” in memory; it constitutes what is generally called *long-term* (*conceptual*) *memory*. The higher the level of *activation* a piece of information has the greater the probability for it to be retrieved from memory to enter the flow of information processing. Activation spreads among chunks along their links and rapidly decays in time, assuring that only a few related pieces of information are retrieved from long-term memory at any given time. This very small set of highly active chunks constitute the content of *short-term* (*conceptual*) *memory*, also referred to as *working memory*.

Information-processing by the human cognitive architecture is regarded as purposeful, with

explicit *goals* represented in memory. Goals can be part of an activity script, a result of sensory input or generated by cognitive processes. Not yet accomplished goals are accessed constantly by ongoing cognitive processes, maintaining a high level of activation and hence kept in working memory. At the same time, active goals spread activation towards other elements in working and long-term memory, especially the script describing how to carry through the ongoing task. This script is interpreted by another built-in mechanism, one that implements means-end analysis, resulting in new subgoals. Whilst some of these subgoals can be accomplished directly, others activate new activity scripts which are interpreted in turn by the same mechanism of means-end analysis. Information processing based on this interpretation of activity schemas can combine sensory input with internal goals in many ways, and the content of working memory at any given time gives it a *focus of attention* that greatly improves efficiency (e.g. there is no need to search the whole content of long-term memory because information is retrieved from memory by means of its association with the focus of attention). In addition, the facility of chunking propositions into complex schemas that work as a unit allows overcoming tight limitations in size and persistence of working memory, giving this form of information processing high flexibility and power.

There are situations though in which flexibility and power are much less important than speed and accuracy. For example, a squash player needs to react very quickly and with accuracy to a fast moving ball; a lorry driver on a long journey requires very little flexibility in his cognitive processes (in a relative sense, of course) to do most of his job, yet controlling a vehicle running at 110 km/h certainly demands high speed and precision to respond to a continually evolving environment. Cognitive processes associated with this sort of tasks have very low demands on attention and working memory resources (e.g. the lorry driver can be talking to a friend, or listening to his favourite radio station while driving), are triggered by patterns in sensory input and are less responsive to voluntary control. In general, human beings rely on a combination of *controlled* and *automatic* processes to undertake most of our duties. Most, if not all, of our everyday activities involve a good deal of automatic processes; for example, walking/cycling/driving from home to the office, reading the newspaper, ringing home and typing at the computer keyboard—see (Ericsson & Lehmann, 1996; Stillings et al., 1995, Ch. 2) for detailed examples of tasks modelled, at least partially, as automatic processes. On the other hand, we require the cognitive flexibility and power of controlled processes to perform tasks as simple as getting change for the bus on an unfamiliar street, and as complex as planning holidays with tight constraints on time and economic resources.

The internal representations responsible for the automatic processes are stored in *proce-*

*dural memory*; they are optimised for pattern-based retrieval and execution free of interpretation. Conceptual and procedural memory differ in other important properties besides their main functionality of storing *know-what* and *know-how*, respectively: information stored in conceptual memory is generally accessible to the central cognitive processes, particularly to consciousness, and adaptable to be used in novel ways; information stored in procedural memory, on the contrary, is mostly inaccessible to consciousness and can only be used in a single way<sup>1</sup> (Anderson, 1993).

The classical view of the human cognitive architecture described above is a simplification that abstracts many details from a number of different and more specific theories, which differ in aspects such as degree of parallelism, activation processes, declarative and procedural memory, conflict resolution, goal-oriented behaviour and mechanism of learning (Anderson & Lebiere, 1998, Ch. 12). A more radical departure from the classical view is the connectionist approach to explaining human cognition, which rejects the assumption that human cognition is essentially symbol processing, puts emphasis on the parallel distributed processing that occurs in the brain and conceives cognition as a type of emergent behaviour (Bechtel & Abrahamsen, 1991; Stillings et al., 1995, Sec. 2.10).

### 2.1.1 Skill acquisition

Traditionally, skills have been classified in two main categories: perceptual-motor and cognitive. *Perceptual-motor skills* demand that the performer perceives, interprets and organises sensory information in order to move with various combinations of accuracy, strength and power (Cratty, 1973). *Cognitive skills*, on the other hand, place their demands on abilities for problem solving in ‘intellectual tasks’ (VanLehn, 1996) with various combinations of accuracy, speed and quality.

Although the distribution of effort among the distinct components of the human cognitive architecture (and the human body) is different for acquiring each type of skill, the acquisition processes show a number of similarities. For example, both perceptual-motor and cognitive skills can be seen as developing in stages (Ericsson & Oliver, 1995; Fitts, 1964; VanLehn, 1996), as follows.

1. The early stage, in which basic knowledge<sup>2</sup> about the task (what it is about, how to

---

<sup>1</sup>The distinctions between conceptual and procedural memory, and between long-term and short-term memory are made here at the functional level only. No claims are made about differences between their location or implementation in the nervous system.

<sup>2</sup>The term ‘knowledge’ is used here in a broad sense, referring to the information stored in both conceptual and procedural memory, and including “proper” knowledge and misconceptions.

accomplish it) is acquired.

2. The intermediate stage, in which arduous practice leads to mastering of the knowledge and flawless performance (free of errors other than occasional slips).
3. The final stage, where slow fine tuning of the skill follows the famous Power Law of Practice, which says that improvement in performance is directly proportional to the amount of practice up to a certain exponent (Newell & Rosenbloom, 1981).

The demands of the skill on working memory and attention decrease with practice (Cratty, 1973; VanLehn, 1996), signalling a shift from dependency on controlled processes and declarative knowledge to dependency on automatic processes, procedural knowledge and the sensori-motor subsystems.

A number of proposals have been put forward of built-in mechanisms underlying the shift from controlled to automated processes in skill acquisition, among them *knowledge compilation*, from conceptual into procedural form (Anderson & Lebiere, 1998), discrimination and generalisation of the patterns that prompt the execution of procedural knowledge (Anderson, 1983), *chunking* of small pieces of knowledge to form bigger structures that are processed more efficiently (Anderson, 1983; Newell, 1990), *strengthening* the associations between relevant pieces of knowledge (Anderson & Lebiere, 1998), adaptations in the perception and early processing of information (Goldstone, 1998; Haider & Frensch, 1996), and physiological and anatomical adjustments (Ericsson & Lehmann, 1996).

### 2.1.2 Meta-cognition

*Meta-cognition* has to do both with knowledge about cognition, our own and of others, and with the regulation of cognition (Brown, 1987; Flavell, 1979, 1987). The only difference between knowledge and *meta-knowledge* is in their subject matter, not in their form or quality (Flavell, 1979). Consequently, acquisition, representation and general processing of meta-knowledge should conform to the description of the human cognitive architecture given above (Section 2.1), and hence it is possible for meta-knowledge to be encoded either in conceptual or procedural form, leading to both controlled and automatic cognitive processes. In the same way, meta-cognitive skills pertain to our abilities to regulate our knowing, learning and understanding.

Examples of meta-knowledge include knowledge about one's own knowledge (e.g. 'I know the basics of structured programming') and cognitive abilities (e.g. 'my memory is very poor').

Examples of meta-cognitive abilities are self-explaining and monitoring of understanding (Van-Lehn, 1996).

## 2.2 Effects of participative learner modelling

Participative learner modelling was characterised in Section 1.1 by an active and explicit involvement of the learner in the modelling process. A comparison was made with more traditional learner modelling in which the learner is a mere object of observation and diagnosis, frequently unaware that this is happening. The function of the educative task attributed to participative learner modelling, as a promoter of learners' reflection and knowledge awareness, means that it should have a positive effect on learners' knowledge and skills; in other words, learners should know "more" or "better" when exposed to participative learner modelling than otherwise. Furthermore, these effects on learners' cognition should be explainable in terms of the human cognitive architecture (Section 2.1) and testable in practice.

In order to simplify the tasks of explaining and testing the effects of participative learner modelling, it is convenient to focus the analysis to the case where the role of the learner in the modelling process is limited to inspecting the content of the learner model, and to modifying it if he judges necessary. The narrowing of the scope of the research in this way should not limit its generality, because an inspectable learner model is a feature shared by most of the forms of participative learner modelling described in Section 1.4 (see also Section 1.7).

Inspecting (and modifying) a learner model in particular, and participating in learner modelling in general, are tasks that call for meta-cognition, as it is emphasized by the frequent use of the words 'reflection' and 'awareness' in the literature (e.g. Bull et al., 1995; Bull & Shurville, 1999; Kay, 1997; Paiva et al., 1995). Compared with direct instruction by the system or exploration by the learner of a detached subject matter like geometry, language, or how to fix a radio receiver, inspecting the learner model has the learner's own cognition as the domain of discourse (Self, 1999b).

The image in the mirror can be compared only with the learner's conceptual knowledge. Procedural knowledge is inaccessible to conscious inspection and hence cannot be compared directly with the image, although a comparison may be made with memories of past performances, with elaborations on these memories (e.g. rationalisations) or with conceptual knowledge left over from earlier stages in the acquisition of a skill (Anderson, 1993).

The effects on the learner of undertaking this comparison can be of many different kinds. One possibility is that the learner acquires new conceptual knowledge about the domain. When



	<u>Your confidence:</u> (a–d)	<u>System confidence:</u> (1–4)
<u>The pronoun is:</u>		
pre-verbal in negatives e.g. Não os compra	• unsure (c)	• very sure (1)
post-verbal in positive main clauses e.g. Compra-os	• almost sure (b)	• unsure (3)

**Figure 2.2:** Example of a component of the learner model in MR COLLINS. Adapted with permission from (Bull, 1997a).

learning Portuguese as a second language with MR COLLINS (Bull, 1997a), for example, the learner may get the composite proposition ‘the pronoun is pre-verbal in negatives and post-verbal in positive main clauses’ represented in his declarative memory after he sees the component of the learner model shown in Figure 2.2. If a representation of the proposition already exists in the learner’s declarative memory, then its level of activation is increased (Section 2.1; Anderson, 1993). This is part of the language awareness that Bull expected MR COLLINS to raise.

A second possibility is that the learner reinforces his existing meta-cognitive knowledge, or increases it. As an example of knowledge of what Flavell (1987) calls *task variables*, the identification of knowledge components marked in the learner model as unknown to a user of the SAM editor may lead this user to a better appreciation of the extent of his knowledge of the editor (Cook & Kay, 1994). Cook & Kay give also an example of knowledge of *strategy variables*: they report that one of the participants in their studies commented that the learner model had helped him to identify elements at the frontier of his knowledge and he used this self-knowledge to guide his learning. Another example is provided by Bull & Ma (submitted) who report that one of the participants in their experiments became more aware of her language learning strategies after participating in a learner modelling task. An example of knowledge of *person variables* is given by Beck et al. (1997), who comment on preliminary evidence of an increase in self-confidence among the (female) participants in their experiments.

The existence of new conceptual knowledge and meta-knowledge, or its increased activation, should be verifiable through changes in the behaviour of learners which occur after they have inspected (and possibly modified) their learner models. The effects should be particularly strong in learning environments for developing skills that rely heavily on the acquisition of procedural knowledge (highly efficient for fast execution but otherwise very constrained).

The reason for this to be so is that, gradually with practice, the original controlled processes that drive unskilled performance get substituted by automatic processes in skilled performance (Section 2.1.1); the retrieval and interpretation of declarative knowledge is substituted by the execution of more efficient procedural knowledge. Consequently, as declarative knowledge gets less used, it becomes less activated and hence less accessible. Under these conditions of low activation of declarative knowledge, or its complete removal from memory, participative learner modelling has the opportunity to introduce a more significant change in learner's cognition by reactivating or re-creating declarative knowledge. In other conditions (i.e. active declarative knowledge) the change introduced by participative learner modelling would be less dramatic.

Among the changes in learner behaviour that can be expected after inspecting a learner model are:

- increased ability to articulate accurate domain knowledge, due to the creation and/or activation of declarative knowledge and meta-knowledge (task variables);
- increased ability to use domain (declarative) knowledge in flexible ways, particularly on novel tasks and conditions; and
- an initial decrease in performance of the original skill, due to interference of declarative knowledge (new or reactivated) with more efficient procedural knowledge—this is possible because knowledge is retrieved stochastically from memory (procedural or declarative) with a probability that is directly proportional to its level of activation (Section 2.1).

Whereas the previous paragraph presents a conclusion derived from a theory of the human cognitive architecture and an analysis of the task of inspecting the learner model, the changes listed above are working hypotheses that can be tested empirically, once they are instantiated by adding the details of a particular domain.

## 2.3 Dependency on the theory

The proposed effects of participative learner modelling on the cognitive state and behaviour of the learner depend in varying degrees on the theoretical assumptions about the human cognitive architecture, skill acquisition and metacognition outlined in the previous sections. Other theories may produce a different set of predictions. For example, the suggested explanation of the effects of participative learner modelling in terms of the acquisition of conceptual knowledge

and meta-knowledge seems incompatible with a connectionist proposal for the human cognitive architecture—yet a recent special issue of *Applied Intelligence*, summarised by Kurfess (1999), is devoted to knowledge representation and reasoning using neural networks.

The distinction between *explicit* and *implicit* knowledge is different from the distinction between conceptual and procedural knowledge outlined in Section 2.1, although they are closely related (Berry & Broadbent, 1984; Dienes & Perner, 1999; Schacter, 1999; Stillings et al., 1995, Sec. 3.2). Explicit knowledge of past events can be recollected or reported; implicit knowledge cannot, yet its existence is detectable through changes it produces on behaviour. Research has indicated that conceptual knowledge can be implicit, and that might be the case for knowledge acquired in participative learner modelling, which learners might acquire and yet being incapable of reporting it.

The suggestion that the effects of participative learner modelling should be stronger in domains that promote the acquisition of procedural knowledge is undermined by theories that propose restructuring of knowledge and not its compilation as an alternative explanation of skill acquisition (Cheng, 1985). It has been shown also that the reportability, accessibility, flexibility and transfer of knowledge acquired through practice in problem-solving domains depend very much on the context in which the task is performed, such as the how the task is presented, the instructions given, the salience of the rules governing the task and the goals pursued (Berry & Broadbent, 1984, 1988; Geddes & Stevenson, 1997). Participative learner modelling may also interact with these factors—in a similar way as goals and salience were shown interacting by Berry & Broadbent (1988)—producing different effects.

The assumption that compilation of declarative knowledge into procedural form occurs during skill acquisition is not essential to the argument that lead to the effects of participative learner modelling hypothesised above. Consequently, the hypothesis are not affected by research that indicates procedural knowledge can be acquired independently of declarative knowledge (e.g. Berry & Broadbent, 1984). On the other hand, other researchers have developed theories and computational models of cognitive architectures capable of bottom-up skill acquisition, which start learning procedural knowledge and develop conceptual knowledge later (Karmiloff-Smith, 1992, 1996; Sun et al., 1996, 1999). From the viewpoint of these theories, participative learner modelling may have the potential of contributing to the naturally occurring development of conceptual knowledge during skill acquisition, without interfering with previously learnt procedural knowledge. However, questions may arise as to whether participative learner modelling is anchored enough in the learner's experience with the original task for it to have a positive and long-lasting effect on the learner's knowledge.

The conclusion that participative learner modelling should lead to the acquisition of new or the reactivation of already existing conceptual knowledge and meta-knowledge, together with the idea that its effects should be particularly strong in domains demanding the acquisition of perceptual-motor skills, are the principles that guide the investigation of participative learner modelling described in the following chapters. In particular, they justify the selection of a domain centred on a perceptual-motor control task for further exploring of the effects on learners of inspecting their learner model.

The next chapter describes a domain suitable for testing the hypotheses about changes in learner behaviour enumerated at the end of the previous section. However, the instantiation of these hypotheses and the report of the experiments conducted for testing them is postponed until Chapter 7. In between there is the description of PACMOD: the system that implements inspectable learner models in the chosen domain.

## Chapter 3

# A testing domain: balancing a pole on a cart

Interactive computer games ... are the application that will soon need human-level AI, and they can provide the environments for research on the right kind of problems that lead to the type of the incremental and integrative research needed to achieve human-level AI.

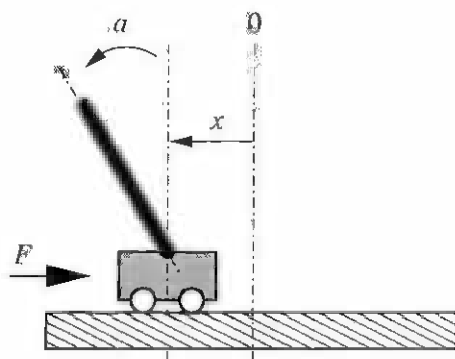
J. E. Laird and M. van Lent (2000). *Human-level AI's killer application: interactive computer games.*

Testing the effects on learners of inspecting their learner models requires the construction of learner models in the first place, and modelling learners demands a domain to be learnt. So this chapter introduces the task of balancing a pole on a cart, justifies why it is a good testing domain for constructing inspectable learner models and includes a review of relevant work done on this and other related tasks.

### 3.1 The pole and cart problem

Most people have tried at least once to stand a stick or an inverted broom upright on the palm of a hand or on a finger. This task, like riding a bicycle, juggling balls, playing the computer game of TETRIS or driving a car, demands fast and accurate motor reaction to (at least apparently) continuous changes in the state of some part of the environment—e.g. the stick, the path ahead, the balls, the falling tile on the screen, and the traffic—monitored by the performer using her senses, specially her vision. In other words, performing the task demands a fair amount of perceptual-motor skills (Section 2.1.1; Annett, 1995; Fitts, 1964).

A classical problem in the theory of control, very similar to stand a stick on a finger, is that of balancing a rigid pole hinged to the top of a cart which is mounted on a straight rail of finite length (Eastwood, 1968; Figure 3.1). In this case, the pole can swing only on the vertical plane passing through the rail on which the cart runs; that is to say, if the cart can move from left to right and from right to left, then the pole can swing clockwise and anti-clockwise only. The whole device can be controlled solely by the application (or not) of a horizontal force parallel to the movement of the cart. An analysis of the physical and mathematical properties of the pole and cart device, together with the outline of a solution to the problem from control theory, can be found in Appendix A. A variation of the problem, with the extra “bang-bang” condition that the force to be applied has fixed magnitude, has been used as test-bed application in research on machine learning (e.g. Barto et al., 1983; Dasgupta, 1998; Michie & Chambers, 1968; Moriarty & Miikkulainen, 1996; Srinivasan & Camacho, 1999), knowledge acquisition (Michie et al., 1990) and qualitative reasoning (Bratko, 1995; Makarovic, 1991).



**Figure 3.1:** The pole and cart device. A position ( $x$ ) of the cart on the right half of the track is taken as positive; a position on the left half of the track is considered negative. An inclination ( $a$ ) of the pole to the right of the vertical is considered positive; an inclination to the left of the vertical is taken as negative.  $F$  is the external force applied to control the device.

## 3.2 Why balancing balancing a pole on a cart?

Balancing a pole on a cart is not the kind of domain learner modelling has usually been applied to—although there is at least one system that uses the pole and cart problem to convey computer-aided instruction of theory of control (Meixler Technologies, Inc., 1999). The sensorimotor skill necessary to control the pole and cart contrasts with the more cognitive skills demanded by the sort of domains learner modelling has commonly been implemented for, such

as solving problems in arithmetics or geometry and writing computer programs (e.g. Anderson et al., 1985; Brown & Burton, 1978; Corbett et al., 1990; Johnson, 1990; Webb & Kuzmycz, 1996). Nevertheless, controlling the pole and cart has a number of properties that make it suitable for exploring participative learner modelling.

1. The real time nature of the task prompts learners to monitor constantly the state of the device and to respond quickly and accurately to changes in its state. This property should favour the quick development of implicit procedural knowledge, in detriment of explicit conceptual representations (see Chapter 2, specially end of Section 2.2; Michie et al., 1990). As a consequence, the effects on learners of inspecting their learner models predicted in Section 2.2 should be clear in this context (but see Chapter 7).
2. Controlling the pole and cart It is a well studied problem in Artificial Intelligence, where several solutions are available (e.g. Barto et al., 1983; Bratko, 1995; Dasgupta, 1998; Michie & Chambers, 1968; Moriarty & Miikkulainen, 1996). In particular, previous work on knowledge acquisition for developing expert controllers of the pole and cart can be adapted for learner modelling (Chapter 4).
3. It involves a simple and well defined task that facilitates the design and implementation of experiments.
4. It allows testing the feasibility of the participative approach to learner modelling beyond its traditional domains of application (reviewed in Section 1.4).

A important clarification is due here: there is no claim that participative learner modelling is a useful thing to do for improving novices' controlling skills; the only claim made here is that controlling the pole and cart can be useful for exploring participative learner modelling. That said, expanding learner modelling (whether participative or not) to sensorimotor domains may prove to be useful in the near future. Computer power and input and output devices that yesterday were confined to research laboratories and the military are nowadays finding their way to the desktop. Sophisticated audio and video cards, video-cameras, steering wheels and joysticks with "force feedback" can now be bought in any decent computer shop; eye-tracking and movement-tracking devices are on their way. These technologies extend the type and amount of information that can be exchanged between learners and current intelligent tutoring systems (e.g. Gluck et al., 2000) and are expanding the range of domains *intelligent* learning environments can be implemented for: from driving cars to flying airplanes, from operation of tools to playing a musical instrument and traffic control, from surgery to firefighting. These

domains involve a great deal of sensorimotor skills, and there will be a need to build learner models for them. Tackling a simpler but closely related task, like controlling the pole and cart, can illuminate the road ahead (cf. Laird & van Lent, 2000).

### 3.3 Balancing the pole and cart in PacMod

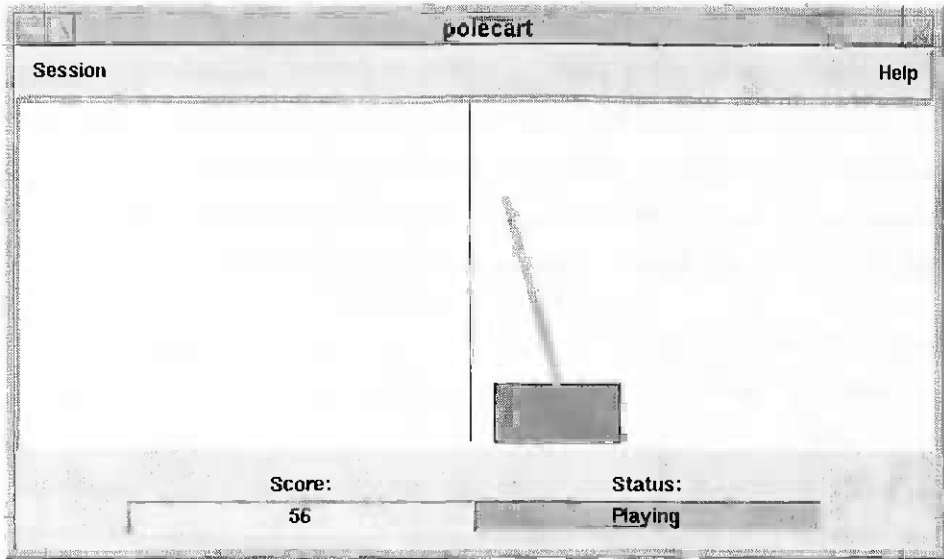
A common practice in most of the research mentioned above is the use of computer-based simulators instead of physical devices. The same practice was followed in this research, using a simulator based on code from Finton (1994) coupled with a graphical user interface (Figure 3.2). A state of the pole and cart device is determined by four variables: the angle ( $\alpha$ ) and angular velocity ( $\dot{\alpha}$ ) of the pole; the position ( $x$ ) and velocity ( $\dot{x}$ ) of the cart. Given a state  $S_i = (\alpha_i, \dot{\alpha}_i, x_i, \dot{x}_i)$  of the device and an external force  $F$ , the simulator calculates a new state  $S_{i+1} = (\alpha_{i+1}, \dot{\alpha}_{i+1}, x_{i+1}, \dot{x}_{i+1})$  using a standard set of values for the physical parameters of the device (Table 3.1) and a time step of 20 ms. Henceforth any reference to the ‘pole and cart (device)’ should be interpreted as being about this simulator; for example, controlling the pole and cart will mean preventing the (simulated) pole from toppling over the (simulated) cart, and preventing the (simulated) cart from leaving the window.

**Table 3.1:** A standard set of parameters for the problem of balancing a pole on a cart (Bratko, 1995). The length of the track corresponds to the maximal distance the centre of the cart (fixed point of the pole) can traverse.

Parameter	Value
Cart mass	1 kg
Pole mass	0.1 kg
Pole length	1 m
Magnitude of force	10 N
Length of track	4.8 m

User input for controlling purposes is restricted to pressing three arrow keys:  $\uparrow$  to start the simulator,  $\leftarrow$  to push the cart to the left, and  $\rightarrow$  to push the cart to the right. Keystrokes are collected for 100 ms and the action corresponding to the last keystroke is sent to the simulator; the simulator calculates the subsequent state of the device, which is displayed in the graphical window, and the cycle starts again. The simulation is five times slower than the real physical process because the simulator calculates the state of the device in time steps of 20 ms while the interface sends actions to the simulator every 100 ms.





**Figure 3.2:** Graphical user interface to a simulator of the pole and cart device.

The computer system used to run the simulator is very reliable in its timing<sup>1</sup>: an average duration of 99.98 ms ( $s = 2.12$ ) for the input-simulation-output cycle was obtained after thirteen hours of practice with the simulator summed up in the series of studies comprised by this research (see Chapters 5, 6 and 7 for reports of the studies). The participants input at least one action in 36% of all cycles but more than one action in less than 10.5%. Multiple different actions in a cycle happened in less than 0.15% of all cycles, showing that very few actions are discarded in spite of the fact that only the last user action per interval of 100 ms is considered.

Initially, the auto-repeat-key facility of the operating system was disabled, so that pressing a key and holding it down produced only one action. However, the facility was enabled again after some users commented they found the original configuration unnatural. The advantage of the original configuration is that each keystroke has to be produced explicitly by the user, and so action timing can be seen as a property of user performance. In the second configuration additional “keystrokes” are produced automatically by the system when a user holds down a key, making it more difficult to associate action timing with user traits (but see Section 3.4).

Three informative studies and two experiments constitute the empirical work (with human participants) of this research<sup>2</sup>. In all of them, the participants’ attempts to controlling the pole and cart occurred in a number of uninterrupted *control sessions*, each session consisting of a

<sup>1</sup>The computer is a Sun ULTRA 5 running SOLARIS and X WINDOW as operating system and window manager, respectively.

<sup>2</sup>Chapter 5 contains descriptions of experiments with artificial controllers of the pole and cart.

series of *control runs*. A control run began when a participant pressed  $\uparrow$  to start the simulator; it included the subsequent chain of states of the pole and cart (calculated by the simulator) and actions from the user; and ended when a *crash* occurred (i.e. the cart reached a window border or the pole toppled over the cart) or the user pressed  $\uparrow$  again.

Depending on the study or experiment, the control runs started with states chosen from one of three different sets: easy-centred, semi-random or hard-displaced.

**Easy-centred** initial states are only two, with all state variables—cart position ( $x$ ) and velocity ( $\dot{x}$ ); pole angle ( $a$ ) and angular velocity ( $\dot{a}$ )—set to zero, except the angle of the pole which is set to  $\pm 6^\circ$  ( $\pm 0.1048$  radians). In other words, the cart appears centred and immobile on the window, whereas the pole appears still but slightly tilted either to the right or to the left. These states represent rather easy starting conditions to control.

**Hard-displaced** initial states are also two, but they have all variables set to nonzero values, as follows:  $x = \pm 0.4915$  m,  $\dot{x} = \pm 0.7150$  m/s,  $a = \pm 0.1054$  rad and  $\dot{a} = \pm 0.2968$  rad/s; all values chosen with the same sign. In words, the cart appears displaced, not far from the centre of the window but moving towards the border; the pole appears slightly tilted and falling in the same direction the cart is displaced. Contrary to easy-centred states, hard-displaced states pose rather difficult starting conditions to control.

**Semi-random** initial states are constructed in a way that makes their degree of difficulty as starting conditions to control to vary from very easy to rather difficult. A threshold was defined for each one of the state variables, based on the preconditions of the rules used by the artificial controllers described in Chapter 5:  $x_t = \pm 0.5$  m,  $\dot{x}_t = \pm 0.4$  m/s,  $a_t = \pm 0.07$  rad and  $\dot{a}_t = \pm 0.5$  rad/s. Semi-random states are constructed by randomly selecting values for the variables  $\dot{a}$ ,  $a$ ,  $\dot{x}$  and  $x$  (in that order) so that at most one of these values lied outside the interval  $[-v_t, +v_t]$  but still inside the interval  $[-2v_t, +2v_t]$ , where  $v$  stands for the corresponding variable. The idea behind this procedure is that any state has at most one variable with a value that makes control more difficult (e.g. the cart can be quite displaced but not moving fast; the pole can be moving quickly but close to the vertical).

Easy-centred states were used in the early stages of the research, but it was found that they make the task too easy, producing samples of controlling behaviour that are too narrow to reliably infer a broader pattern of behaviour from them (Morales & Pain, 1999). The variability among semi-random states allows better sampling of behaviour that benefits learner modelling, but

makes comparing performance between individual learners more difficult. The consistency of hard-displaced states (and easy-centred states too) is better for comparing performance between individual learners—see Chapter 5 for more details of the impact of choosing initial states from any of the three sets on the modelling of human novices and artificial controllers.

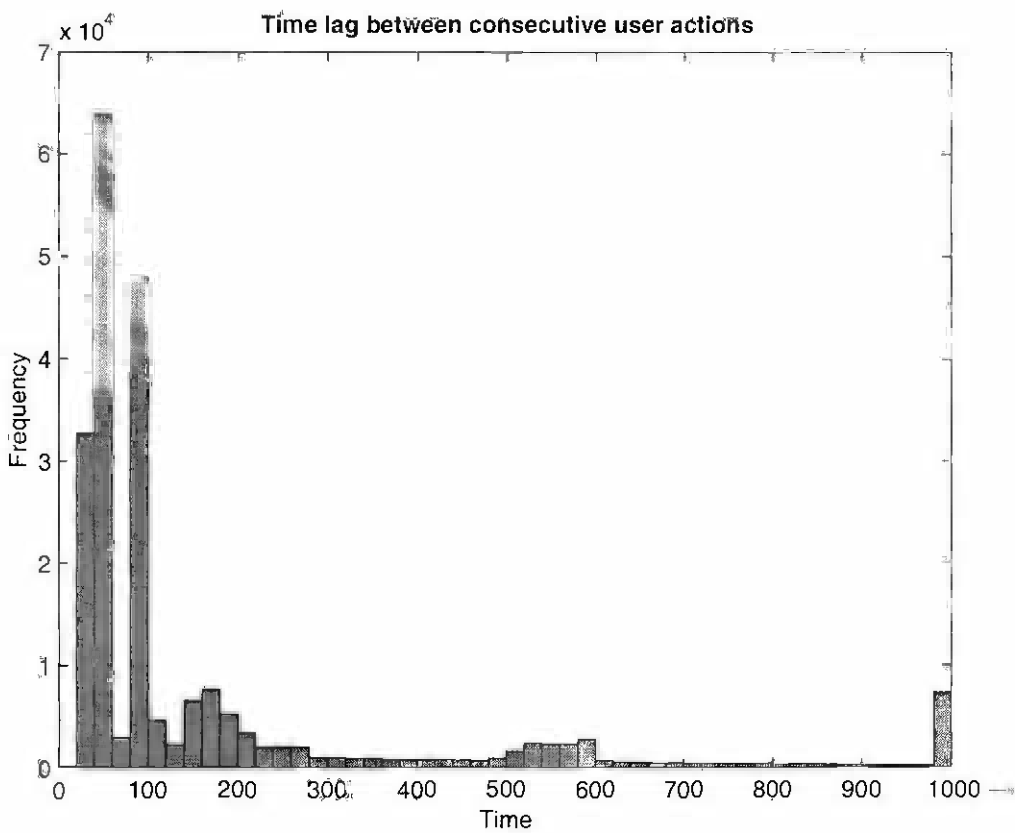
### 3.4 Human control of the pole and cart

There are several factors that affect the way a person learns and performs a task like balancing a pole on a cart. First of all, there are capabilities and limitations inherent in the human cognitive architecture (Section 2.1) that constrain any person's approach to the task, as well as her readiness, accuracy and consistency. Secondly, each person contributes with her own perceptual, cognitive and motor abilities for sensing the states of the device, recognising relevant features, selecting appropriate actions and executing them through pressing the corresponding keys. The strategic knowledge employed by a person in pursuing the task, encoded either declaratively or procedurally and executed by a mixture of controlled and automatic process, is an important component of that person's way of performing the task. All these factors are closely interrelated. For example, the way strategic knowledge is represented affects the speed with which it can be processed, leading to variations in the delay from perceiving states to issuing actions; however, a person can use her knowledge to plan a sequence of actions which can then be executed timely, reducing the effect on performance of perceptual and motor limitations. Finally, there are stochastic fluctuations in cognitive processing, environmental noise, slips, etc., which introduce random variations in the mapping between states of the pole and cart and a person's actions.

According to Michie (1998) and Hayes-Michie (1999), experiments on learning to control the pole and cart revealed a large variation in learning times and gained expertise among individuals, with gender, age and educational background identified as important factors affecting learning speed and performance. In a prior paper, Michie et al. (1990) reported a study in which people became noticeably better at the task after less than half an hour of total practice, judged from a reduction in the number of failures to keep the pole and cart under control. They observed that subjects paid particular attention to the pole, resulting in a clear reduction in the number of failures due to dropping the pole between the first and the last tests. On the contrary, the number of failures due to running off the edge remained essentially constant.

Psychological theories of how humans perform perceptual-motor tasks (e.g. Byrne & Anderson, 1998; Meyer & Kieras, 1997a,b, which also contains a review) predict that control of

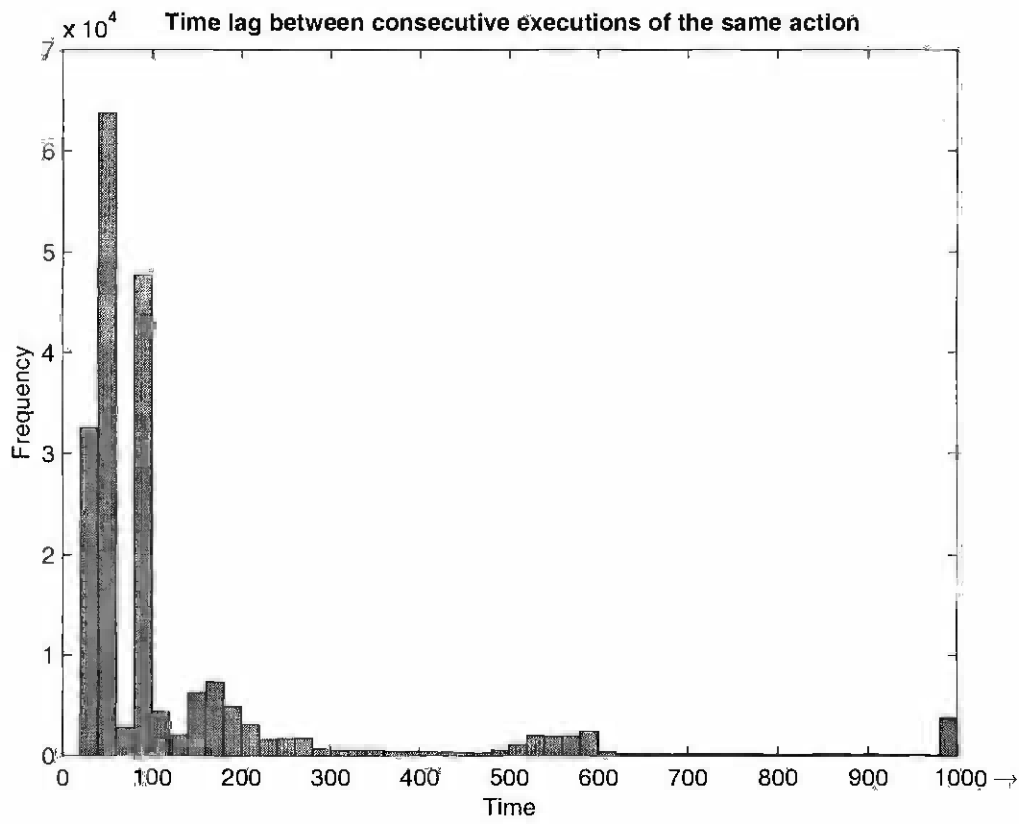
the pole and cart by humans will be intermittent, because it results from a discrete sequence of cycles involving perception, cognition and action. The serial aspects of human cognition limit both the minimum length for each cycle (a *reaction-time* of around 300 ms) and the amount of overlapping between cycles. The theory is supported by results from experiments on the task of visual-manual tracking (Poulton, 1966). For example, an average time lag of 500 ms between consecutive actions has been found for early stages of visual-manual tracking, with typical values ranging from 250 ms to one second (Vince, 1948, as cited in Poulton, 1966, p. 389). Studies of human performance in TETRIS, a video-game very popular in the 1980's, have shown that time lags between actions vary significantly from a mean of near 400 ms in beginners to a mean of near 300 ms in experts (Maglio, 1995, Ch. 3).



**Figure 3.3:** Histogram that illustrates the frequency distribution of time lag between consecutive user actions. The bar on the far right represents time lags from about one second up to a maximum of 12.4 s.

Figure 3.3 shows a histogram illustrating the frequency distribution of time lag between consecutive actions (keypresses) executed by the participants in the studies and experiments.

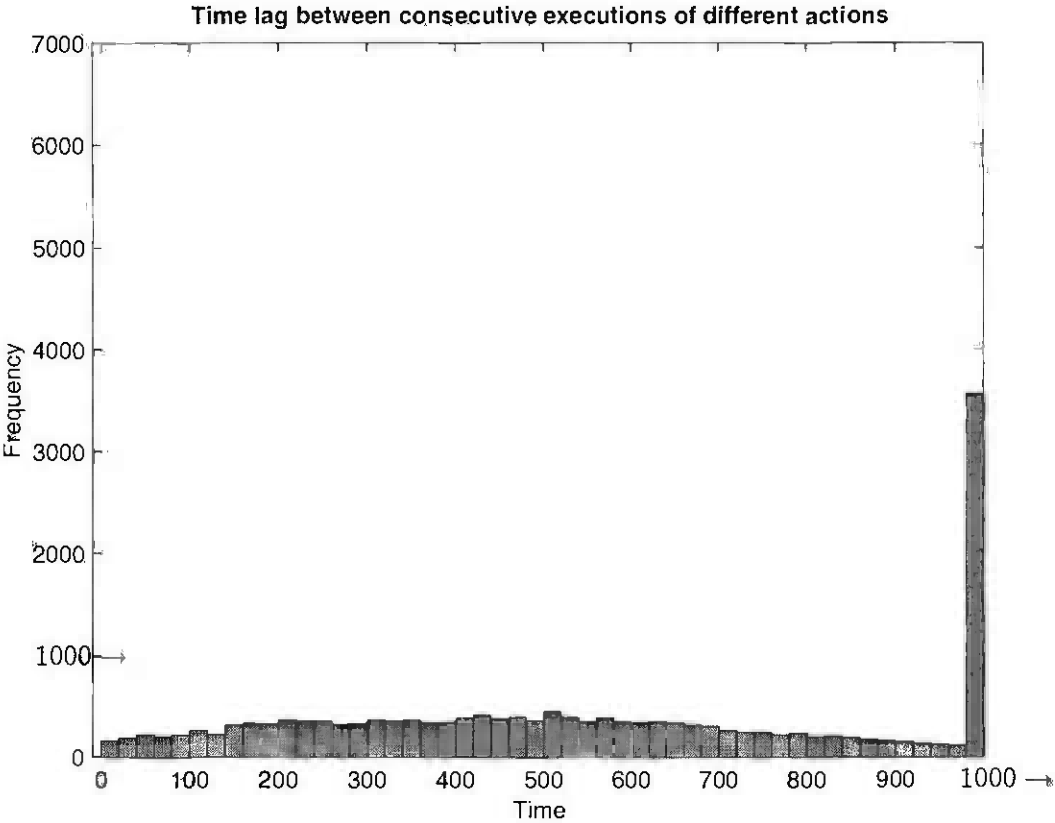
mentioned in Section 3.1. The high peaks near 50 ms and 100 ms result from the repeat-key facility provided by the operating system: they correspond to the threshold for the system to start auto-repeating a keypress (100 ms) and the time lag between the automatic repetitions (50 ms). The remaining data suggests two “typical” time lags between actions around 180 ms and 550 ms. A tentative interpretation of the smaller value is that the continuous nature of the task facilitates overlapping of perception/cognition/action cycles—as it happens also in experiments on the psychological refractory period (Meyer & Kieras, 1997a). The smaller value is also smaller than typical time lags found in visual-manual tracking and TETRIS. This fact can be attributed to the amount of information provided by the graphical interface (specially compared to the simpler screens normally used in visual-manual tracking) and the user accumulating knowledge of the task (more predictable than TETRIS).



**Figure 3.4:** Histogram that illustrates the frequency distribution of time lag between consecutive executions of the same action. The bar on the far right represents time lags from about one second up to a maximum of 12.4 s.

The shape of the distribution is dictated to a great extent by the time lags between con-

secutive executions of the same action (Figure 3.4). The reason is simple: there are far fewer consecutive executions of different actions and the distribution of their time lag is flatter (Figure 3.5). This fact indicates human performance of the task is characterised by burst of identical actions. The shortest and most common time lags between consecutive actions are due to the timing of the repeat-key facility provided by the operating system ( $\approx 50$  ms and  $\approx 100$  ms). Longer time lags centres around either 180 ms or 550 ms. The peak near 180 ms can be interpreted as the user being able to predict future changes in the pole and cart, planning a sequence of (identical) actions in advance, and either wanting more control over action timing than allowed by the repeat-key facility or being unable to rely on this facility (as it happened in the original configuration of PACMOD, described in Section 3.3). The peak near 550 ms can be attributed more easily to reaction time.



**Figure 3.5:** Histogram that illustrates the frequency distribution of time lag between consecutive executions of different actions. The bar on the far right represents time lags from about one second up to a maximum of 8.3 s.

## Chapter 4

# Construction of learner models

Our approach is different: having the semantic network as an information structure on the subject being dealt with, it seems natural to consider it as an input-output model of the ideal student. It is so to the extent that the semantic network, when interrogated, would give the same answers a “perfect” student would. In other words, we are not claiming that a perfect student has his knowledge organised strictly the way the semantic network is organised. We simply claim that both would produce, when interrogated, essentially the same output.

Jaime R. Carbonell (1970). *AI in CAI: An artificial intelligence approach to computer assisted instruction.*

This chapter contains a description of the learner modelling method used by PACMOD (the umbrella name for the collection of programs that implement inspectable learner models in the domain of controlling the pole and cart). It begins with a swift overview of learner modelling that puts PACMOD’s method in context before describing the method in detail. The chapter contains also short reviews of machine learning for student<sup>1</sup> modelling and of ‘behavioural cloning’, a technique for extracting expert knowledge from expert behaviour on which PACMOD’s method of diagnosis is based.

### 4.1 Options for learner modelling

Given a subject domain, there are many approaches to learner modelling to choose from—see (VanLehn, 1988) and (Wenger, 1987, Ch. 16–17) for reviews of “classical” formulations of the learner modelling problem and methods for tackling it; a selection of newer approaches

---

<sup>1</sup>The term ‘student’ is used frequently in this chapter to avoid confusion between human learners and machine learners.

and techniques can be found in (Greer & McCalla, 1994). However, the characteristics of a particular domain frequently suggest that some ways of modelling its learners are more appropriate than others. For example, domains in which special emphasis is on problem-solving skills, such as basic arithmetic and programming, may hint using a procedural knowledge representation—e.g. BUGGY's procedural network (Brown & Burton, 1978; Wenger, 1987, Ch. 8) and the system of production rules of the LISP Tutor (Reiser et al., 1985). Other domains, in which conceptual learning is more important, such as geography and meteorology, may suggest the use of a more declarative knowledge representation—e.g. a semantic network in SCHOLAR (Carbonell, 1970) and a hierarchy of *scripts* in WHY (Stevens et al., 1982); see also (Wenger, 1987, Ch. 3)<sup>2</sup>.

Learner modelling techniques have been classified in many different ways, according to aspects such as the sort of knowledge represented in the models, the knowledge representation employed, the relation between learner models and expert knowledge, the way information about the learner is collected, the method of diagnosis that is used, the kind of inferences derived from observations of learner behaviour, the degree to which each learner model is tailored to its target learner, and so on (VanLehn, 1988; Wenger, 1987, Ch. 16–17). Two classifications deserve special attention here: knowledge-based versus interaction-driven learner modelling, and behavioural versus epistemic learner modelling (Wenger, 1987, Ch. 17). The reason for selecting these two classification schemes is that the learner modelling method implemented by PACMOD (the Pole And Cart learner MODeller; see Sections 4.1.3 and 4.4) can be categorised as both behavioural and interaction-driven, in a subcategory of behavioural modelling that Wenger (1987, Ch. 11) calls *post-hoc reconstructive interpretation*. Other systems in the same category are PROUST (Johnson, 1990) and the MACSYMA advisor (Genesereth, 1982).

#### 4.1.1 Knowledge-based vs interaction-driven

*Knowledge-based* learner modelling constructs models in terms of detailed formulations of domain knowledge, common variations on them regarded as “faulty” or plain wrong (and hence called misconceptions, mal-rules or bugs) and theories of learning. Learner models are maintained on the basis of comparisons between the behaviour of the learners and the behaviour expected from an expert (Clancey, 1983) or an ‘ideal’ or ‘perfect’ student (Carbonell, 1970;

---

<sup>2</sup>The distinction between *procedural* and *declarative* knowledge representation is clear and subtle at the same time. It is related to efficiency of processing and flexibility of use of a representation, but also depends on the capabilities of the processor and its faculties to access the representation. For more on the declarative/procedural distinction, from the viewpoint of student modelling, see (VanLehn, 1988). See (Winograd, 1975) for a view from artificial intelligence, and Section 2.1 for a view from cognitive psychology.



Reiser et al., 1985). This form of learner modelling has often been criticised (e.g. Elsom-Cook, 1993) for oversimplifying the differences between learners and experts, and for demanding the elaboration of a detailed inventory of domain knowledge, bugs and misconceptions. In addition, a single formulation of domain knowledge makes it difficult to accommodate for differences in how individual learners approach the subject matter. In contrast, *interaction-driven* learner modelling requires less prior knowledge, relying more on data gathered in the course of system-user interaction. In a sense, it regards learners as individuals developing their own skills and personal understanding of the domain. Representative of this approach are methods that focus on modelling the ‘course of interaction’, as opposed to modelling the learners’ knowledge state (Akhras & Self, 1997; Self, 1999a), and methods that employ machine learning techniques to analyse learner behaviour and produce generalisations that explain it (Section 4.2).

#### 4.1.2 Behavioural vs epistemic

*Behavioural* learner modelling tries to evaluate, characterise, predict or reproduce learner behaviour, with a minimum of inferences about the learner’s internal knowledge state. Wenger (1987) presents WEST (Burton & Brown, 1982), GUIDON (Clancey, 1983) and PROUST (Johnson, 1990) as typical examples of behavioural learner modelling. *Epistemic* learner modelling, on the contrary, regards behaviour as the symptom of an internal knowledge state which it tries to infer. Classical examples of systems that employ this sort of learner modelling are those based on Anderson’s theories of human cognition (Anderson et al., 1990). Wenger introduces the term ‘unobservable behaviour’ to denote the ‘use of knowledge in a reasoning chain’ (a sort of middle point in between behaviour and knowledge) and argues that some instances of behavioural modelling aim to infer the former. He concedes, nevertheless, that the distinction ‘can be subtle and may only make sense in the context of a communication environment’. The fact that he classifies the learner modelling method employed by WEST both as behavioural and epistemic shows that there is no clear cut between the categories<sup>3</sup>.

#### 4.1.3 Learner modelling for the pole and cart

PACMOD creates each learner model from scratch, based on a relatively long sequence of input-output records of the form (*device state, learner action*). Constructing a learner model

<sup>3</sup>A third kind (or level) of modelling is considered also by Wenger, which he calls *individual* learner modelling: it has to do with other aspects of learners besides their knowledge state, such as their learning style, (type of) personality, motivational state, context, intentions, meta-cognitive aspects, and their model of the tutoring system (as opposed to the system’s model of the learner).

from this sort of data can be viewed as a classification task; the resulting model is a theory that generalises over the collection of records (training data), in the sense that it permits to ascribe a learner action to every possible state of the pole and cart—hopefully the action the learner would most probably execute for each case. From this perspective, the application of classification techniques from machine learning (Section 4.2) to modelling the student looks quite attractive and straightforward. The method of diagnosis used by PACMOD is an adaptation of a technique for automatic acquisition of expert knowledge called ‘behavioural cloning’ (Section 4.3), which employs a machine-learning technique called ‘supervised rule induction’ (Mitchell, 1997, Ch. 10) to extract expert knowledge from traces of expert behaviour.

PACMOD produces sets of rules as student models. Each rule specifies a pattern of values of the state variables that prompt the student to execute one of the three available actions: *push-left*, *push-right* or *wait* (the “action” of doing nothing)—see example in Figure 4.4. The use of rule-based learner models is driven by the goal of constructing models that are comprehensible to students, and it is supported by previous research on student modelling (e.g. Clancey, 1983), inspectable student models (Bull & Pain, 1995) and expert systems (Hayes-Roth, 1985).

Because machine learning does not necessarily relate to human learning, it cannot be claimed that students build in their minds a theory as described in their student models. In fact, only a handful of assumptions about control of the pole and cart and the way humans learn to do it are made for modelling the students. This is not because knowledge about the domain and learning in the domain are useless; on the contrary, much research in the area testifies it can be very useful. A different matter is whether this knowledge is necessary for achieving the goal of constructing student models that students can view as clear descriptions of the strategies they follow when solving a certain kind of problems; models that students are willing to identify themselves with. Consequently, although PACMOD’s student models are *predictive*, they are *explanatory* only to the extent that they express regularities in student behaviour, and there is no intention of them representing internal structures or processes of human students (cf. Lee, 1999).

## 4.2 Machine learning

Using machine learning for student modelling is appealing and has a long history; after all, the idea of improving the performance of computer programs through learning from experience (Mitchell, 1997) sounds very attractive. In a detailed review published recently, Sison & Shimura (1998), distinguish between two main uses of machine learning techniques for stu-

dent modelling: to construct student models, and to improve the performance of the student modelling component of the system. In the first case, the system learns the student model from interactions with the student, and hence becomes better at supporting *his* learning. In the second case, the system makes use of previous modelling experiences to improve on *its* ability to construct “good” student models. Hereafter this section deals with the former case, since that is where PACMOD’s method fits—see (Sison & Shimura, *ibid.*) for more details on the latter.

In order to construct a student model, a computer system can try to guess the effects of tutoring interactions on the student’s knowledge. One way of doing it is by modelling human learning on artificial learning (Ohlsson, 1992; Self, 1986); another way is by employing machine learning techniques to interpret student behaviour (Gilmore & Self, 1988; Langley et al., 1984; Sleeman, 1982; Webb & Kuzmycz, 1996). In the latter case, machine learning offers the possibility of student modelling that is led by the actual behaviour of the student and does not require a detailed description of the domain knowledge and its “faulty” deviations (Elsom-Cook, 1993; Wenger, 1987, s. 10.4); in other words, interaction-driven student modelling<sup>4</sup> (Section 4.1.1). Assuming less about the student and his domain knowledge gives grounds for expecting a decrease in the bias of the diagnosis; it can introduce greater flexibility to accommodate different (human) learning styles and conceptualisations of the domain (Jonassen & Grabowski, 1993; Wenger’s notion of ‘viewpoints’, *ibid.*). In the concrete case of controlling the pole and cart, different control styles can be diagnosed and modelled: from a cautious style that attempts to stabilise the device to a venturesome balancing that to and fro between the extremes of the window.

The use of machine learning to diagnose a student’s knowledge from his behaviour is not straightforward. On the contrary, it poses very interesting and difficult problems, among which Sison & Shimura put emphasis on two:

- i) inconsistencies in student behaviour produce an unusually high amount of noise in the training data, and
- ii) changes in student behaviour (specially due to their awaited learning of the domain) make student knowledge a moving target for machine learning—a phenomenon known in the area as *concept drift* (Schlimmer & Granger, 1986).

In fact, Sison & Shimura are not very optimistic about the prospects of this use of machine learning: they regard the mobility of the target ‘considerably higher than what most machine

<sup>4</sup>Nevertheless, machine learning can be used also for finding the holes for an *overlay model*, and for creating, detecting and selecting bugs for a *perturbation model*—two prototypical types of knowledge-based student models (Section 4.1.1). See (Sison & Shimura, *ibid.*) for details.

learning systems normally deal with', and guess that 'the kind and degree of noise in student modelling might be too much for existing inductive learning techniques to handle' (Ibid., p. 40). From this perspective, the method of student modelling described latter in this chapter acquires a particular significance, because it shows how in some cases the problem of noise—and, to a lesser extent, changes in student behaviour—can be handled.

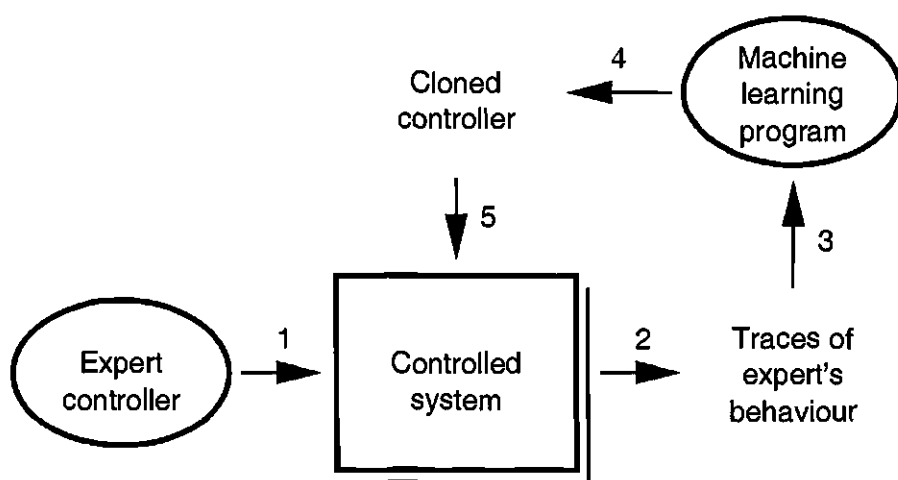
A number of alternative machine learning techniques are available: rule discovery, decision trees, Bayesian networks, neural networks and case-based reasoning are some of them (Mitchell, 1997). Every technique will impose a particular structure upon the resulting model, and although it is sometimes relatively easy to convert from one representation into another (e.g. from decision trees to production rules) such a conversion is often very difficult and cumbersome, if feasible at all (e.g. from neural networks to production rules). In this research a decisive criterion for selecting a particular technique was the production of models structured in a way that facilitates their inspection, understanding, and possible modification by apprentice controllers of the pole and cart who may be unfamiliar with techniques for knowledge representation (Chapter 6). Production rules are then considered a good choice because they match the nature of the task and have been proved effective for modelling human skill acquisition and performance (Anderson, 1993; Anderson & Lebiere, 1998); they are also easy to interpret in operational terms, have a symbolic character and simple structure that resembles familiar rules of thumb and support modularity of representation (Barr & Feigenbaum, 1981; Hayes-Roth, 1985). Nevertheless, this is a design decision; other knowledge representation techniques could be equally suitable—for example, Zapata-Rivera & Greer (2000) show how student models based on Bayesian networks can be opened to student inspection.

As commented on in Section 4.1.3, the models do not necessarily describe plausible internal processes or internal representations of human students. This view is shared by other researchers in the area of machine learning applied to student modelling (Gilmore & Self, 1988; Webb & Kuzmycz, 1996) in relation to the models built by their systems, but it is not the only one. Langley et al. (1984), for example, embed the machine-learning techniques in a broader theory of cognition, and so may claim some degree of psychological plausibility.

### 4.3 Behavioural cloning

The term 'behavioural cloning' was coined by Donald Michie to denote a method of acquiring expert knowledge from traces of expert behaviour by means of machine learning (Michie, 1993; cited in Bratko et al., 1997). The method has been applied to the tasks of controlling the

pole and cart (Michie et al., 1990), flying a plane (Camacho, 1998; Michie & Camacho, 1994), operating a crane (Urbančič & Bratko, 1994) and production scheduling (Kerr & Kibira, 1994; cited in Bratko et al., *ibid.*)— see (Bratko et al., *ibid.*) for a review of behavioural cloning applied to these domains. The development of the method is motivated by the difficulties encountered in getting expert performers of sensorimotor control tasks to produce accurate explanations of their skills that can be translated into computer systems. Expert control skills rely heavily on perceptual and motor adaptations, as well as on automatic processes that result from the execution of knowledge encoded in procedural form; explanations of the skills, on the other hand, are generated from conceptual knowledge and rationalisations of memories of past events (Sections 2.1 and Section 2.1.1). Consequently, it is natural to encounter inconsistencies between what experts say and do.



**Figure 4.1:** The method of behavioural cloning.

Broadly speaking, the method consists of five steps (Figure 4.1).

1. A (human) expert performs the target task in the environment (real or simulated in a computer) under a variety of conditions.
2. Traces of the expert's behaviour are recorded, containing information about values of selected attributes of the system being controlled (e.g. velocity of the cart, altitude of the plane, angle of the crane's rope, buffer levels in the production line) and corresponding expert actions.
3. A machine learning program takes the traces of behaviour as training data and produces a generalisation of it, from which "expert actions" can be inferred for all possible states

of the controlled system.

4. The output of the machine learning program is incorporated into an expert system.
5. The performance of the system at the task is evaluated, and any necessary adjustments are done.

It has been observed in all applications of behavioural cloning that the “controlling style” of the resulting controller—understood as the relevant qualitative features of the way the system is controlled—resembles the style of the expert it has being derived from: the new controller is, in this sense, a “clone” of the original one; hence the name of the method. Nevertheless, the method is able to extract a polished version of the original expert’s control strategy that allows the clone to generally outperform the original expert—a phenomenon that has become known as the ‘clean-up effect’ of behavioural cloning. In order to get a robust clone, on the other hand, it is very important to get the original expert to perform the target task under a variety of conditions that is representative of the range of possible scenarios of control; otherwise, there is a high risk of ending with a clone that performs like the original expert only in a narrow range of conditions.

A slightly secondary issue has been to fine tune the training data in order to associate each expert action precisely with the state of the controlled system that elicited it. In their review, Bratko et al. (1997) suggest exploration to find the match that produces best results, starting with assuming no delay. On the one hand, both cognitive limitations that produce time lags between perception and action and the cognitive mechanisms available to compensate them have to be taken into account to decide which state of the system triggered each action. On the other hand, the natural continuity of many control tasks and the clean-up effect mentioned above seem to greatly compensate the lack of accuracy in the matching between system states and expert actions.

Bratko et al. notice that some important information about the task is missing in simple traces of behaviour. For example, there is no distinction between critical actions and less important ones, nor between actions produced by expertise and actions that result from mistakes and slips. While the clean-up effect shows that machine learning is capable of dealing with part of the noise introduced by mistakes and slips, Bratko et al. suggest the use of background domain knowledge to estimate the relative importance of distinct expert actions. The key use of background knowledge suggested by Bratko et al. is nevertheless to give a deeper structure to the clone’s knowledge, because up to now they have lacked ‘the conceptual structure typical in human control strategies: goals and subgoals, phases and causality’ (ibid., p. 349). Van Lent

& Laird (1999) provide an interesting example of how background knowledge can be incorporated into the process: by presenting experts with a goal hierarchy and requesting them to annotate their actions with the goal they have in mind. A simple algorithm of supervised learning is then used to produce expert knowledge encoded in terms of SOAR's language of problem spaces, operators and goals. Van Lent & Laird have tried this approach in two distinct domains (tactical air combat and the video-game *QUAKE II*) but only to clone predefined artificial controllers (i.e. a faultless *annotator*). It is an open question whether a similar approach would work with human experts.

#### 4.4 From traces of behaviour to learner models

The fact that behavioural cloning applied to a control task can produce 'an articulate account of the given acquired skill' (Michie et al., 1990, p. 75) leads naturally to the question of whether the same method can be used for learner modelling. The shift of target from experts to apprentices makes this application of behavioural cloning different to previous ones in two key aspects:

1. The goal is to reproduce a learner's knowledge, not to maximise the clone's expertise; so the clone should approximate as close as possible the controlling behaviour of the learner rather than surpass it.
2. The behaviour of a beginner exhibits more faults and inconsistencies than the behaviour of an expert, and it is also expected to change over time. This situation might put the levels of noise and concept drift far beyond the power of inductive machine learning (see Section 4.2).

A framework for constructing learner models out of traces of learner behaviour is outlined in Figure 4.2. It consists of four stages: (1) recording of learner behaviour, (2) preparation of the traces for diagnosis, (3) induction of a set of production rules using machine learning and (4) informed refinement of the set of rules to produce the final learner model. This framework can be applied to a wide range of tasks: from tasks executed under tight time constraints, of which controlling the pole and cart is an example, to more slowly paced tasks such as programming or writing in a second language (cf. Section 4.5). The remainder of this section contains a description of the instantiation of this framework in PACMOD to constructing models of novice controllers of the pole and cart (Chapter 3).

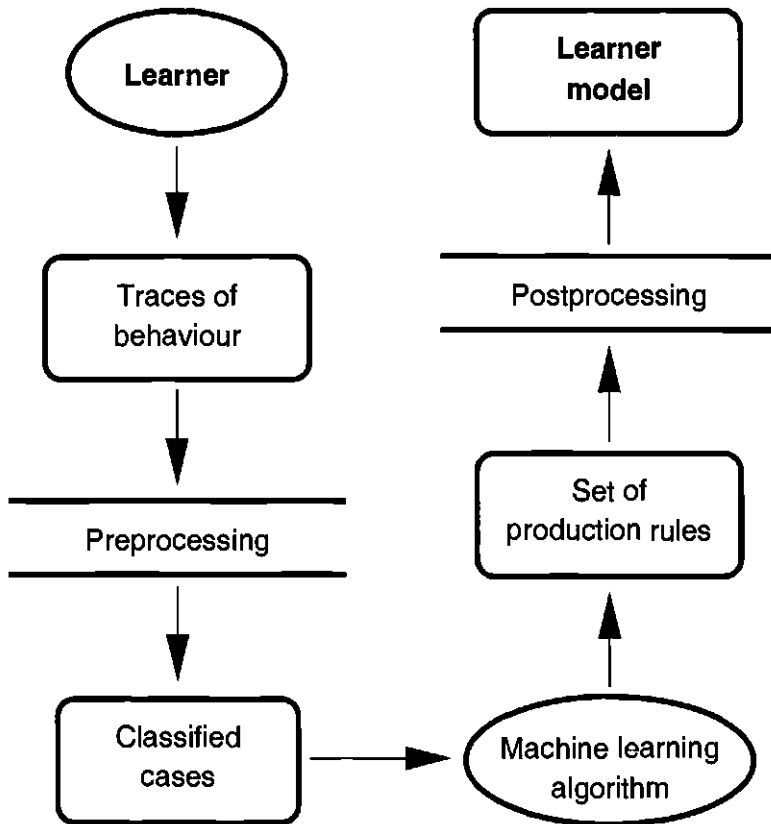


Figure 4.2: A framework for constructing learner models from traces of learner behaviour.

#### 4.4.1 Traces of behaviour

Controlling the pole and cart with PACMOD can occur in a number of *sessions*, each one consisting of a series of *control runs*. A control run begins when the learner presses  $\uparrow$  to start the simulator; it includes the subsequent chain of states of the pole and cart (calculated by the simulator) and actions from the learner; and ends when a *crash* occurs (i.e. the cart has fallen off at one end of the rail or the pole has reached a horizontal position) or the learner presses  $\uparrow$  again. PACMOD associates each learner action with the state of the pole and cart current at the moment the action was executed and saves each control run in a separate file containing records of the form  $(device\ state, action)$ , where

*device state* is a vector of the form  $(a, \dot{a}, x, \dot{x})$ , with pole angles ( $a$ ) measured in radians pole angular velocities ( $\dot{a}$ ) in radians per second, cart positions ( $x$ ) in metres and cart velocities ( $\dot{x}$ ) in metres per second; and



*learner action* is any of *push-left*, *push-right* or *wait* (see also Section 3.3)<sup>5</sup>.

Velocities correspond to the instantaneous velocities as calculated by the simulator, which are assumed to be perceived by learners as directly as they perceive cart position and pole angle.

#### 4.4.2 Pre-processing

The traces of behaviour described above have one characteristic that make them inappropriate as direct input to a standard algorithm for machine learning: a certain degree of misalignment between device states and student actions, and the nature and amount of noise they contain. In addition, it is useful to give the learning algorithm information about the symmetry in the task, since this property seems evident to humans—even if there are a number of reasons for the actual performance of the task not being symmetrical: the  $\leftarrow$  and  $\rightarrow$  keys are not placed in a symmetrical position on the keyboard (they are both on the left-hand side of it); the window that displays the state of the pole and cart may be placed asymmetrically on the screen; the learner may employ different fingers to press the  $\rightarrow$  and  $\leftarrow$  keys, which are not symmetrically located in his body; he may perceive differently left movement and right movement, etc. The intention here is to reduce the possibility of confusing the learners by presenting them rather asymmetrical learner models.

Some pre-processing of the collection of control run traces is hence in order; the details of it are given below. A final set of records is produced as a result which describes better the correspondence between states of the pole and cart and learner actions. These records are the raw material for generating a model of the strategy employed by the learner to control the pole and cart.

##### Misalignment due to reaction time

The state of the pole and cart changes over time even if the learner does not carry out any action. This fact and the existence of time lags between the perception of a state and the execution of an action (Section 3.4) make improbable that a given action corresponds exactly to the state of the device it coincides with. The correct alignment of states and actions (or a good approximation to it) has to be found before applying any machine-learning algorithm.

Finding a good alignment of states and actions happens to be a difficult task for a number of reasons. Firstly, although we all share physical limitations that greatly determine our reaction

---

<sup>5</sup>Each record contains also the following additional information: the time it was recorded, a counter, and the full list of learner actions collected in the last 100 ms interval. See the end of Section 3.1 for an example of use of this information.

speed to stimuli, every one of us exhibits individual sensorimotor traits that make us depart more or less from the mean. Secondly, sensorimotor limitations can be overcome using higher-level cognitive skills involving knowledge, reasoning and planning. For example, when the pole falling at high speed elicits burst of identical actions from a learner, it is easy to interpret his behaviour as produced by his knowledge that it is necessary to repeat the same action several times in order to achieve a more stable condition, and then planning to do that. On the other hand, a very stable condition (the pole and cart moving at very low speed) gives learners the opportunity to evaluate the current situation, predict when a qualitative change would take place, and then prepare in advance the appropriate action for it. Finally, although it seems possible to infer a good approximation to the correct alignment of states and actions by identifying states of the pole and cart for which there are obvious actions to execute, and hence using them as hints to the correct alignment of traces, that move has a serious implication: it presupposes the existence of “normal behaviour” against which the behaviour of learners can be measured, casting the shadow of traditional overlay learner modelling over an otherwise data-driven and learner-centred approach (Section 4.1.3).

Research on visual-manual tracking and the game of TETRIS (Section 3.4) suggests reaction times in between 300 and 500 milliseconds. Michie et al. (1990) opt for a fixed time delay of 400 ms for cloning an expert on the pole and cart problem, yet they conclude it should have been longer. Bratko et al. (1997) suggest that is questionable to apply classical results of reaction time to sudden stimuli to ongoing tasks like controlling a dynamic system, and propose a trial and error approach starting with zero delay.

In principle, PACMOD uses a fixed reaction delay of 300 ms for all learners; a choice loosely based on time lags between consecutive actions observed in the first two studies of this research (Section 3.4). However, since the state of the pole and cart is updated only every 100 ms and the action corresponding to the last keystroke in the previous 100 ms is the only one taken into account (see Section 3.3 for details), the actual reaction delay used by PACMOD fluctuates between 200 ms and 400 ms. Other values may be as good as these ones, or better, but these produced acceptable results (Chapter 5) and it is unclear why any others could do better.

### Noisy data

Due to reaction time, the fact that a learner does not execute any action for a certain period of time does not necessarily reflect his decision to do so; on the contrary, it may simply be the case that he is not fast enough to provide an action during that period, even if he is willing

to. A perfect alignment of device states and learner actions does not eliminate these bogus *wait* actions: it simply makes them correspond to the state of the device the learner could not respond to with a proper action. So, traces of behaviour contain noise, and the non-randomness of this noise would make a learning algorithm minimise the induction error by accurately classifying noisy *wait* actions (which generally constitute the majority class) in detriment of a good classification of pushing actions.

Michie et al. (1990) solve the problem by recording only pushing actions in the traces; furthermore, they only take into account an action when it is different from the previous one. This solution produces clones that kept pushing the pole and cart all the time, which is not bad if the goal is to clone expertise but it is not appropriate for the purposes of this research.

PACMOD attempts to reduce the amount of noise by removing isolated recordings of *wait* actions from the traces of behaviour, on the grounds that they are more susceptible to being the consequence of reaction delays. In this research, this meant a reduction in the proportion of *wait* actions in the training data from 64% to 30% (the total training data was reduced by 49%).

### Symmetry of the domain

A minimum amount of knowledge about the domain of controlling the pole and cart is implicit in the data input into the learning algorithm: the number of state variables, their types and ranges, and the set of available actions. Knowledge about more specific traits of the domain, such as symmetry and the relationships between variables, and between variables and actions, are implicit in the distribution of the data for the learning algorithm to extract and make explicit.

Because the symmetry of the task is considered obvious to human controllers<sup>6</sup>, it is convenient for their models to reflect this belief. Two ways of increasing the symmetry of the models were considered: either to introduce symmetrical cases in the input to the learning algorithm, or to add symmetrical rules to its output. In the first case, for every existing record  $(\pm x, \pm \dot{x}, \pm a, \pm \dot{a}) \rightarrow action$  in the training data, a new record  $(\mp x, \mp \dot{x}, \mp a, \mp \dot{a}) \rightarrow action'$  with symmetrical device state and opposite learner action is included too (*push-left* and *push-right* are opposites; *wait* is its own opposite). The symmetry of the output is then left to the learning algorithm.

In the second case, for every rule

**if**  $(var_1 \bigcirc_1 value_1) \wedge \dots \wedge (var_n \bigcirc_n value_n)$  **then** *action*

---

<sup>6</sup>Empirical evidence for this assumption is the comments from several participants in the studies, and the self-reports gathered in Experiment 1a (Section 7.7).

a new rule

$$\text{if } (var_1 \bigcirc'_1 -value_1) \wedge \cdots \wedge (var_n \bigcirc'_n -value_n) \text{ then } action'$$

is added, where  $\bigcirc_i$  stands for a relation of order (say,  $\geq$  or  $\leq$ ),  $\bigcirc'_i$  denotes the opposite relation (i.e.  $\leq$  and  $\geq$ ), and  $action'$  denotes the action opposite to  $action$ , as before.

The first option has the advantage of providing additional information to the induction process, and the disadvantage of only loose control over the symmetry of the output (see Sections 5.2.3 and 5.3.3 for examples of learner models and an estimation of their symmetry). The second option guarantees the symmetry of the result, but it has the disadvantage of increasing the number of rules in the output, and hence the size of the final student model. PACMOD implements the former.

#### 4.4.3 Machine learning program

There are three options when pondering on using machine learning for student modelling: either to develop a home-made technique, to implement an already known one, or to use an existing implementation of the latter. The home-made is the more flexible of all three approaches: it allows advantage to be taken of knowledge about the distinctive features of student modelling in general and the specific application in particular—see (Sison & Shimura, 1998; Webb & Kuzmycz, 1996) for some examples. On the other hand, it can easily be the least cost-effective option, with little or no extra benefits (Webb et al., 1997). Implementing a well-known machine learning technique takes advantage of experience accumulated among the machine learning community; the assets and pitfalls of the technique will be known and documented, and there will be solid basis to justify its application to the problem at hand—and perhaps more places to publish the results! Furthermore, this approach retains most of the flexibility of the home-made one, particularly in terms of adapting the technique to the oddities of the training data and output requirements. Finally, using an existing implementation of a well-known machine learning technique, either in the form of a library—e.g. MLC++ (Kohavi, 1996; Kohavi et al., 1997)—or as a full program—e.g. C4.5 (Quinlan, 1993), CN2 (Clark & Niblett, 1989), OC1 (Murthy et al., 1994)—can be the most cost-effective way of solving a student modelling problem, at the same time as most of the benefits of re-implementing the same technique are preserved. Nevertheless, it might require re-casting a student modelling problem as a machine learning problem, and translating training data and results to and from the particular conventions of an existing tool.

PACMOD uses RIPPER (Cohen, 1995), an existing implementation of an algorithm for supervised rule induction. Informal testing with C4.5 showed only small differences in error rates between the two programs, and RIPPER was chosen because it allows more control of the induction process and its final outcome. RIPPER is instructed to produce rules with a minimum coverage of 1% of the training data, to separate each class (action) from the rest, and to consider false negatives costlier than false positives (to better recover scarce pushing actions from among a majority of *wait* actions).

#### 4.4.4 Post-processing

Figure 4.3 presents a set of production rules as produced by RIPPER. Because RIPPER was told to separate each class (action) from the rest, the result contains rules for every action. The preconditions of the rules are not disjoint, which means that the preconditions of more than one rule can match a given state of the pole and cart (a property discussed further in Chapter 6). In order to solve conflicts, RIPPER gives each rules a “goodness” score using the formula

$$\frac{p+1}{p+n+2} \quad (4.1)$$

where  $p$  and  $n$  are the number of records in the training data that are correctly and incorrectly classified by the rule, respectively; then the rule with the highest score that matches a state of the pole and cart is the one used to predict learner action for that state.

It was mentioned in Section 4.4.2 that *wait* is generally the most frequent action in the training data to RIPPER. As a result of that, and also because RIPPER is told to regard false positives as less costly, the rules with *wait* action normally get the highest scores and dominate prediction. Furthermore, rules with *wait* actions increase the size of the student models, making them potentially more difficult to understand by the students. In order to avoid these problems, the action of the default rule (the last rule in the set) is always set to *wait*, and all other rules with a *wait* action or score lower than the score of the default rule are removed. The formula for calculating the score is also changed from Equation (4.1) to

$$\frac{p+1}{p+n+3} \quad (4.2)$$

to reflect the fact that there are three actions to choose from (Good, 1965). The result of all this post-processing is the final learner model. An example, corresponding to the rule set in Figure 4.3, can be seen in Figure 4.4.

right	163	20	if $\dot{x} \leq -1.454$ and $a \geq -0.029$ .
left	164	24	if $\dot{x} \geq 1.446$ and $a \leq 0.030$ .
left	163	24	if $\dot{x} \geq 1.280$ and $a \leq 0.017$ .
right	383	76	if $x \leq 0.852$ and $a \geq 0.145$ .
left	383	83	if $x \geq -0.794$ and $a \leq -0.073$ and $\dot{a} \leq -0.022$ .
left	440	110	if $x \geq -1.214$ and $a \leq -0.132$ .
right	159	43	if $-1.086 \leq x \leq 1.592$ and $a \geq -0.098$ and $0.330 \leq \dot{a} \leq 0.973$ .
right	328	92	if $x \leq 1.931$ and $\dot{x} \leq 1.079$ and $a \geq -0.100$ and $\dot{a} \geq 0.274$ .
right	378	121	if $a \geq 0.120$ and $\dot{a} \leq 0.892$ .
left	325	110	if $x \geq -2.163$ and $\dot{x} \geq -1.426$ and $a \leq 0.118$ and $\dot{a} \leq -0.348$ .
wait	125	48	if $-0.710 \leq \dot{x} \leq 1.279$ and $-0.124 \leq a \leq 0.147$ and $\dot{a} \leq 0.029$ .
wait	52	20	if $1.094 \leq x \leq 1.844$ and $a \leq 0.031$ and $\dot{a} \geq 0.082$ .
wait	58	26	if $x \leq -1.990$ and $\dot{a} \leq -0.308$ .
wait	60	27	if $x \geq 1.971$ and $\dot{a} \geq 0.308$ .
wait	216	118	if $-1.279 \leq \dot{x} \leq 0.710$ and $-0.147 \leq a \leq 0.147$ and $\dot{a} \geq -0.285$ .
right	44	77	.
use_best_rule			

**Figure 4.3:** Example of a set of rules induced by RIPPER (Cohen, 1995) for controlling the pole and cart. Variables and their units are described in Section 4.4.1. The two integers in between each rule's action and the rule's preconditions are the number of cases correctly and incorrectly classified by that rule, respectively.

## 4.5 The nature of the models

This chapter has presented a method for constructing learner models in the domain of controlling the pole and cart. Essentially, the method is based on a simple trick: take a technique for acquiring knowledge from experts and apply it to learners instead. The question is whether the trick would work. Learners are not mini-experts nor incomplete-experts nor faulty-experts: they structure and process their knowledge differently to experts. The next chapter is devoted to providing an answer to this question. Meantime, some remarks are pertinent about the nature of characteristics of the modelling process and the nature of its outcome.

Learner modelling on the lines described in this chapter requires minimum knowledge of the domain to be implemented and a minimum of assumptions about the learner. There is no need of an expert model nor an "ideal-learner" model to guide (and constrain) the search for

```

right  163   20  if  $\dot{x} \leq -1.454$  and  $a \geq -0.029$  .
left   164   24  if  $\dot{x} \geq 1.446$  and  $a \leq 0.030$  .
left   163   24  if  $\dot{x} \geq 1.280$  and  $a \leq 0.017$  .
right  383   76  if  $x \leq 0.852$  and  $a \geq 0.145$  .
left   383   83  if  $x \geq -0.794$  and  $a \leq -0.073$  and  $\dot{a} \leq -0.022$  .
left   440  110  if  $x \geq -1.214$  and  $a \leq -0.132$  .
right  159   43  if  $-1.086 \leq x \leq 1.592$  and  $a \geq -0.098$  and  $0.330 \leq \dot{a} \leq 0.973$  .
right  328   92  if  $x \leq 1.931$  and  $\dot{x} \leq 1.079$  and  $a \geq -0.100$  and  $\dot{a} \geq 0.274$  .
right  378  121  if  $a \geq 0.120$  and  $\dot{a} \leq 0.892$  .
left   325  110  if  $x \geq -2.163$  and  $\dot{x} \geq -1.426$  and  $a \leq 0.118$  and  $\dot{a} \leq -0.348$  .
wait    0    0  .
use_best_rule

```

**Figure 4.4:** The final learner model corresponding to the set of rules shown in Figure 4.3.

a suitable learner model. The model is not built in a vacuum, though. Decisions had to be taken on the relevant features for describing learner behaviour and the structure of the learner models; the diagnosis process, both in the preparation of the training data and the search for patterns in it, makes decisions about what is information and what is noise.

The learner models are not only fitted to the behaviour of the learner but they are also justifiable. It is possible to say how much of the original data, from which the learner model was inferred, justifies each one of the rules, and the rules' error rates—this aspect of the models is further discussed in Chapter 6. More details could be given if judged necessary: when the data was gathered, the state of the device and the corresponding learner action, and even a replay of the learner around a specific time (cf. Kay, 2000).

This form of learner modelling has to have its drawbacks. An important one is that it produces a sort of “disposable” learner model. Each model is produced from scratch, based solely on a low-level description of previous learner behaviour. In a real setting, this description cannot go beyond the last few minutes of learner-system interaction before it becomes a burden. A previous learner model would provide a summary of early learner behaviour which could be used to guide the interpretation of the recent events. As it stands, PACMOD would need to go through the old data and redo the analysis; even worse—perhaps (cf. Webb et al., 1997)—it will not distinguish between old and recent behaviour. Another problem with the method is a certain amount of arbitrariness in the models. They are descriptions of patterns in behaviour

with no underlying theory. A different learning algorithm—or maybe the same algorithm with a different random number generator, or a different random seed—may spot different patterns, hence producing quite different models.

Despite the disadvantages, the models produced by PACMOD are suitable for the purposes of this research for three reasons:

1. They are predictive of learner behaviour, a property that not only facilitates their evaluation (Chapter 5) but also makes them useful tools for the analysis of learner behaviour (Chapter 7).
2. They make it possible to go on exploring the possibilities and problems of making them inspectable by learners (Chapter 6).
3. Their disposable nature does not interfere with experimenting on the effects on learners of inspecting and modifying their learner models (Chapter 7)—after all, disposable equipment is ubiquitous in research.



## Chapter 5

# Evaluation of the learner models

There is surely nothing on earth that is completely unrelated to everything else ...If one applies a large enough sample to the study of any relation, trivial or meaningless as it may be, sooner or later a significant result will almost certainly be achieved.

William L. Hays (1994). *Statistics*.

This chapters presents two evaluations of the learner modelling method used by PACMOD. The evaluations focus on the predicting power of the resulting models, their individualised character and PACMOD's accuracy to reproduce the rules directing the behaviour of a set of artificial controllers.

### 5.1 Evaluation of predictive models

Probably the most salient feature of the learner models produced by PACMOD is that they can predict learner behaviour. Given a state of the pole and cart, it is possible to find the rule with the highest score and preconditions match the state: the action of this rule is the model's prediction for the action the learner would execute in the same circumstances. Moreover, the resulting action can be sent to the simulator, as if issued by the learner, leading to a new state of the pole and cart for the model to predict a learner action for. In this way it is possible to execute the learner models to imitate learner behaviour (Collins & Brown, 1988). If individual learners behave differently then modelling individuality becomes a requirement for achieving more accurate predictions and imitation of learner behaviour, since a model tailored to a particular learner would predict better that learner's behaviour than anyone else's.

### 5.1.1 On-line vs off-line prediction

In principle, prediction which takes device states and learner actions from pre-processed traces of behaviour should produce the best results, simply because both training and test data would be prepared in the same way. The result will be a measure of *off-line predicting power*. Nevertheless, it is not clear how much off-line prediction has to say on the capacity of the models to predict learner behaviour in real-time, or *on-line predicting power*: the power to predict the next user action during an interactive session.

The main problem with on-line prediction is to devise a way of handling on-line reaction delays such that they do not penalise prediction too much. A quick and dirty method used in the evaluations below is based on assuming maximum overlap between consecutive perception/cognition/action cycles until contrary evidence is found; then minimum overlap is assumed between the current and immediately following cycle. In other words, if the model predicts a *push-left* or *push-right* action but the learner does nothing (a *wait* action) a reaction delay is assumed and the prediction for the next state is that the learner will do nothing again (another *wait* action). More formally, if  $T = ((s_0, a_0), (s_1, a_1), (s_2, a_2), \dots, (s_n, a_n))$  is a raw trace<sup>1</sup> (i.e. without corrections of reaction delay) with device states  $s_i$  and corresponding user actions  $a_i$ ,  $M$  is a model of the user and  $M(s) = a$  denotes the action the model  $M$  predicts for state  $s$ , then the sequence of predicted actions  $a'_0, a'_1, a'_2, \dots, a'_n$  is defined as follows:

- i)  $a'_0 = a'_1 = a'_2 = \text{wait}$ .
- ii)  $a'_3 = M(s_0)$ .
- iii) For  $j = 4, 5, \dots, n$ , if  $a_{j-1} = \text{wait}$  and  $a'_{j-1} \neq \text{wait}$  then  $a'_j = \text{wait}$  else  $a'_j = M(s_{j-3})$ .

### 5.1.2 Predicting crucial vs predicting non-crucial actions

Not all actions are equal. For example, some *wait* actions are intended by the learner, whereas others are result of reaction delays. One can imagine some situations (states) encountered when controlling the pole and cart are critical, in the sense that individual actions issued then have a significant effect on overall performance and controlling style. On the other hand, actions executed under less critical conditions have little impact on the overall performance. Actions in critical situations should not be predicted incorrectly, whereas actions in less critical situations can be predicted with less accuracy without casting serious doubts on the fidelity of a learner model. Unfortunately, information on whether a situation is more or less critical is missing in

---

<sup>1</sup>On-line prediction can be simulated using raw traces of learner behaviour.

simple traces of behaviour (Bratko et al., 1997; Section 4.3) and hence it could not be used in the evaluation.

### 5.1.3 Evaluation of contents vs evaluation of performance

Comparing the contents of two models is a more complex process than comparing their predictions. This is so because the same learner behaviour can be encoded in many different ways as a set of production rules, and a set of production rules can be arranged in different ways, resulting in different behaviours. However, the fact that the same algorithm is used to build all learner models should produce a sort of implicit standardisation that can be exploited for comparing the contents of the models<sup>2</sup>. This is done in the evaluations only to compare the models of the artificial controllers, as constructed by PACMOD, with the original set of rules that produced their behaviour.

## 5.2 Evaluation of models of artificial controllers

If there were a scanner to see the knowledge used by a human player to control the pole and cart, in the same way X-ray scanners provide visualisations of our internal parts, it could be possible to compare this knowledge with the rules produced by PACMOD<sup>3</sup>. Such a scanner does not exist, but it is possible to build artificial controllers that follow predefined strategies and then compare these strategies with the models induced from traces of the controllers' behaviour.

A family of six such artificial controllers was built, all using the same strategy but with different delays for executing their actions with different levels of error (noise). The strategy employed by these controllers (Table 5.1) is based on the strategy used by Michie et al. (1990) for their artificial controller; the sole difference is in the threshold for the position of the cart, which has been changed from zero to  $\pm 0.5$ . The reason for this modification was to make the controllers more akin to their more relaxed human counterparts (Section 3.4); otherwise, the artificial controllers would keep executing *push-left* and *push-right* but not *wait* actions.

Two of the controllers, henceforth referred to as  $\mathcal{F}_{300,0}$  and  $\mathcal{F}_{300,10}$ , were programmed to delay their actions for 300 ms— $\mathcal{F}_{300,10}$  has 10% of noise added to its output, as explained later, whereas  $\mathcal{F}_{300,0}$  does not have any. The other four controllers had their delays calculated using the formulas

$$RT = (200 + x) \text{ ms} \quad (5.1)$$

<sup>2</sup>This remark was made by one of the reviewers of (Morales & Pain, 1999).

<sup>3</sup>If they are comparable somehow. See Sections 4.2 and 4.5.

**Table 5.1:** The set of rules used by the artificial controllers to control the pole and cart. The strategy behind the rules is to control the pole first and the cart second, giving priority to the velocities. Interestingly, if everything else is under control (i.e. the variables have small values) but the cart is not close to the centre, then this strategy dictates pushing it towards the border!

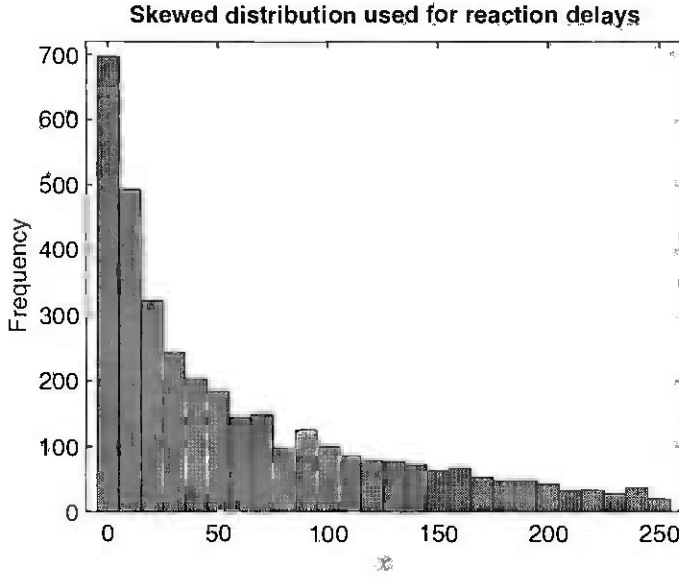
1	wait	10	1	if $-0.500 \leq x \leq 0$ and $-0.400 \leq \dot{x} \leq 0$ and $-0.070 \leq a \leq 0$ and $-0.500 \leq \dot{a} \leq 0$ .
2	right	9	1	if $x \geq 0.500$ and $\dot{x} \geq -0.400$ and $a \geq -0.070$ and $\dot{a} \geq -0.500$ .
3	right	8	1	if $\dot{x} \geq 0.400$ and $a \geq -0.070$ and $\dot{a} \geq -0.500$ .
4	right	7	1	if $a \geq 0.070$ and $\dot{a} \geq -0.500$ .
5	right	6	1	if $\dot{a} \geq 0.500$ .
6	left	5	1	if $x \leq -0.500$ and $\dot{x} \leq 0.400$ and $a \leq 0.070$ and $\dot{a} \leq 0.500$ .
7	left	4	1	if $\dot{x} \leq -0.400$ and $a \leq 0.070$ and $\dot{a} \leq 0.500$ .
8	left	3	1	if $a \leq -0.070$ and $\dot{a} \leq 0.500$ .
9	left	2	1	if $\dot{a} \leq -0.500$ .
10	wait	1	1	.
use_best_rule				

and

$$RT = (400 - x) \text{ ms}, \quad (5.2)$$

where  $x$  was chosen at random from a distribution skewed positively (Figure 5.1). That means the two controllers that used formula Equation (5.1), hereafter referred to as  $\mathcal{V}'_{250,0}$  and  $\mathcal{V}'_{250,10}$  have execution delays in the range  $[200, 450]$  with a mean of around 250 ms; and the remaining two controllers,  $\mathcal{V}'_{350,0}$  and  $\mathcal{V}'_{350,10}$ , which use formula Equation (5.2) had execution delays in the range  $[150, 400]$  but with a mean of around 350 ms. Noise was added to the output of three of the controllers  $\mathcal{F}_{300,10}$ ,  $\mathcal{V}'_{250,10}$  and  $\mathcal{V}'_{350,10}$  as a random sample of 10% of their actions was changed to a different action also chosen at random.

The controllers were set to control the pole and cart starting with the conditions defined by the three different sets of initial states used in the studies with human controllers; namely, easy-centred, hard-displaced and semi-random initial states (Section 3.3). The performance of all artificial controllers during a test period of five minutes is summarised in Table 5.2 in terms of three measures: total number of crashes, median control-run length and proportion of pole crashes (the pole topples over the cart). Total number of crashes and average control-run length



**Figure 5.1:** Histogram that illustrates the skewed distribution used to calculate reaction delays for the artificial controllers. The distribution is based on the  $\chi^2$  distribution and has the following statistics:  $min = 0$ ,  $median = 36$ ,  $max = 250$ ;  $\bar{x} = 60.24$  and  $s = 63.91$ .

are related by the formula

$$\text{Average control-run length} = \frac{\text{Testing period}}{\text{Total number of crashes}}; \quad (5.3)$$

however, choosing the median instead of the average provides a measure of control-run length that adds some extra information, because the median is less sensitive to extreme values. Total number of crashes is the sum of both pole crashes and cart crashes (the cart reaches a border of the window). The proportion of pole crashes, calculates as

$$\text{Proportion of pole crashes} = \frac{\text{Pole crashes}}{\text{Total number of crashes}}; \quad (5.4)$$

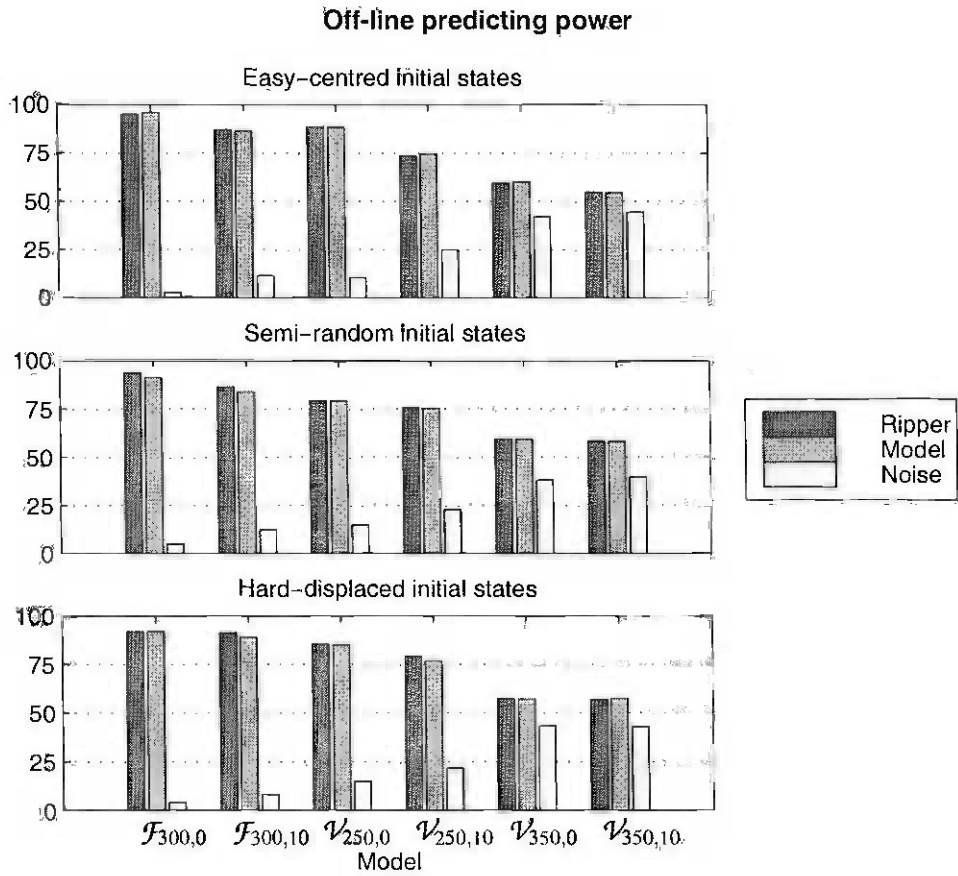
is a measure of the distribution of pole and cart crashes in a control session (see also Section 3.4). As expected, to control the pole and cart starting with a hard-displaced initial state was more difficult than to control them starting with a easy-centred state, with controlling from a semi-random initial states somewhere in between. In fact, the controllers with fixed reaction delay of 300 ms almost never recover from an initial hard-displaced state, suggesting that the ability to react quickly gains in importance when controlling conditions get harder.

Three models for each controller were constructed, one for each set of initial conditions. The performance in the test period of five minutes was used as training data, and the performance in a subsequent period of around a minute was used for testing prediction. The level of

**Table 5.2:** Summary of the performance of the artificial controllers for a test period of five minutes. All control runs that were started during the test period are taken into account, even the ones that ended after that period (one per controller).

Controller	Initial states	Total crashes	Prop. pole crashes	Control-run length				
				Min	1st Q	Median	3rd Q	Max
$\mathcal{F}_{300,0}$	Easy centred	9	1.00	202	254	331	436	468
$\mathcal{F}_{300,10}$		24	0.92	71	100	126	158	192
$\mathcal{V}_{250,0}$		12	0.75	86	182	237	291	519
$\mathcal{V}_{250,10}$		17	1.00	72	105	173	211	309
$\mathcal{V}_{350,0}$		22	0.91	71	79	112	154	335
$\mathcal{V}_{350,10}$		32	1.00	54	72	84	118	183
$\mathcal{F}_{300,0}$	Semi random	23	0.91	52	93	122	164	279
$\mathcal{F}_{300,10}$		31	0.87	51	71.5	76	120.5	268
$\mathcal{V}_{250,0}$		14	0.71	68	90	117	203	861
$\mathcal{V}_{250,10}$		20	0.95	54	81	127.5	183	427
$\mathcal{V}_{350,0}$		27	1.00	47	77	89	149	247
$\mathcal{V}_{350,10}$		38	0.97	42	54	76	91	180
$\mathcal{F}_{300,0}$	Hard displaced	44	1.00	66	68	69	70	70
$\mathcal{F}_{300,10}$		44	0.77	52	54.5	64.5	74.5	148
$\mathcal{V}_{250,0}$		19	0.89	62	90.5	108	149.5	569
$\mathcal{V}_{250,10}$		35	0.77	52	56.5	75	94.5	294
$\mathcal{V}_{350,0}$		45	0.44	53	56	60	76	124
$\mathcal{V}_{350,10}$		50	0.72	40	54	55.5	60.5	107

noise in the test data was measured as the error rate of the exact model (i.e. a model containing the rules shown in Table 5.1); it includes the random noise introduced on purpose plus noise due to misalignment of states and actions because of the controllers’ reaction delays. A similar level of noise can be expected to occur in the training data.



**Figure 5.2:** Off-line predicting power of the models of the artificial controllers and level of noise in testing data.

### 5.2.1 Off-line prediction

The off-line predicting power of the models of the artificial controllers is presented graphically in Figure 5.2, together with the predicting power of the original set of rules generated by RIPPER and the level of noise in the testing data. As expected, the behaviour of the controllers with fixed reaction delay contained the lowest amount of noise and their models were among the more predictive. It can be appreciated also that the output of RIPPER and the final model have in all cases very similar predicting powers. In fact, although paired tests show the difference between their predictions is significant in four cases— $\mathcal{F}_{300,0}$  under semi-random conditions ( $p < 0.025$ );  $\mathcal{F}_{300,10}$  under semi-random and hard-displaced conditions ( $p < 0.025$  and  $p < 0.05$ , respectively); and  $\mathcal{V}_{250,10}$  under hard-displaced conditions ( $p < 0.015$ )—the choice of set of rules accounts for less than 3.1% of the variance, showing that the significance

of the difference can be attributed to the sample size<sup>4</sup> (from 193 to 266 records per controller). In a similar way, paired tests show the difference between predictions by the final models and by the programmed set of rules are significant in six cases— $\mathcal{F}_{300,0}$  under semi-random and hard-displaced conditions ( $p < 0.005$  in both cases);  $\mathcal{F}_{300,10}$  under all conditions ( $p < 0.05$ , 0.005 and 0.03, respectively); and  $\mathcal{V}_{250,0}$  under semi-random conditions ( $p < 0.0001$ )—but the choice of set of rules account for more than 5% of the variance only in two cases:  $\mathcal{F}_{300,0}$  under semi-random and hard conditions (9.4% and 12.6%, respectively).

### 5.2.2 On-line prediction

Figure 5.3 shows the on-line predicting power of the models of the artificial controllers, together with the predicting power of guessing in favour of the most frequent action. The penalty introduced by action delay can be appreciated clearly. For controllers  $\mathcal{F}_{300,x}$  around two thirds of all “actions” were due to delays, of which only half can be treated properly by the algorithm for on-line prediction (Section 5.1.1). For  $\mathcal{V}_{250,x}$ , over half of all actions were consequence of delays, and only a fraction of them can be dealt with by the algorithm. On the contrary, although more than two thirds of all actions attributed to  $\mathcal{V}_{350,x}$  were due to delays, the algorithm or on-line prediction is able to handle more than half of them. Accordingly, models of  $\mathcal{V}_{350,x}$  are the best at on-line predicting and models of  $\mathcal{V}_{250,x}$  are the worst, with models of  $\mathcal{F}_{300,x}$  somewhere in between.

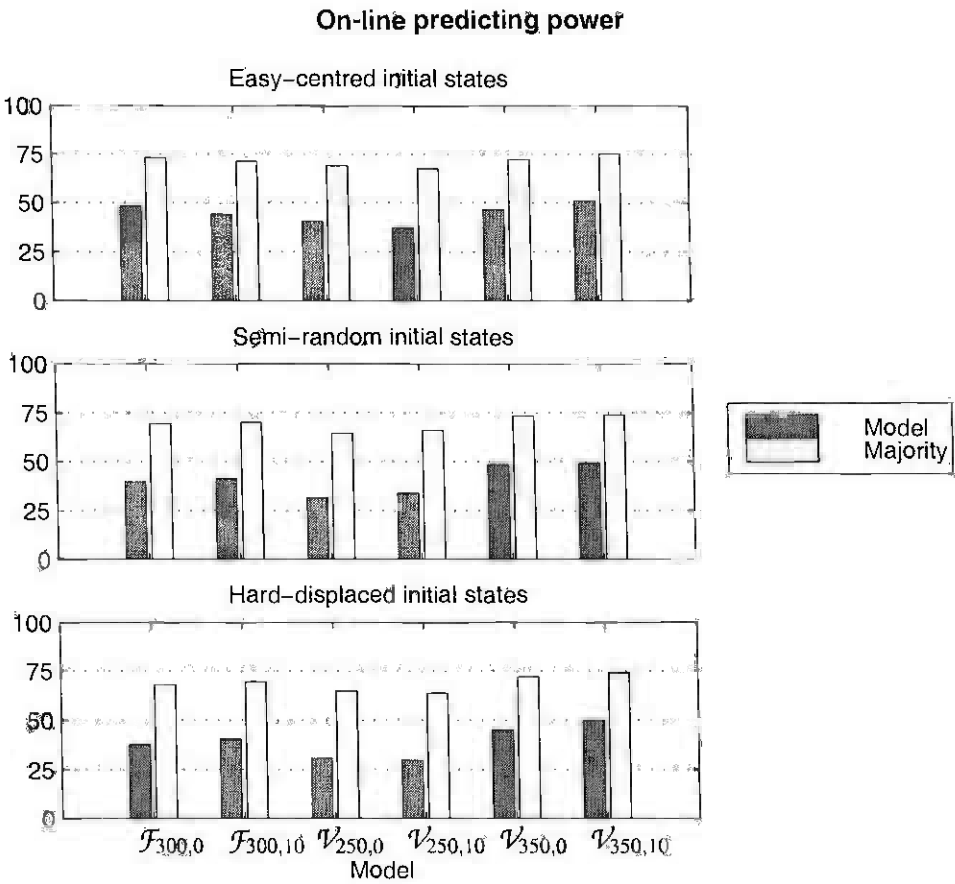
### 5.2.3 Contents of the models

Most of the models of the artificial controllers have either six or seven rules, split more or less evenly between rules for *push-left* and *push-right* actions; the average preconditions per rule is 1.96. A way of estimating the extent to which an induced set of rules resembles the original rules programmed in the artificial controllers is to define a measure of similarity between two arbitrary rules first, and then use it to define a measure of model similarity.

The set of all possible states of the pole and cart can be viewed as contained in a space with four dimensions (a dimension for each one of the state variables: cart position and velocity; pole angle and angular velocity). The subset of the space occupied by the states of the pole and cart is determined by the range of possible values for the position, angle and velocities of the pole and cart. Whereas the position of the cart and the angle of the pole have well defined

<sup>4</sup>A paired test is essentially a *t*-test where the two sets of predictions are interpreted as (related) samples (Weiss & Indurkha, 1998, Ch. 2). The proportion of variance accounted for by the independent variable is discussed in (Hays, 1994, Ch. 8).





**Figure 5.3:** On-line predicting power of the models of the artificial controllers. The predicting power of guessing in favour of the most frequent action (*wait*) is also included.

absolute minimum and maximum values ( $\pm 2.4$  metres and  $\pm 1.571$  radians, respectively), minimum and maximum values for the velocity of the cart and the angular velocity of the pole had to be estimated on the basis of data. They were calculated as  $\pm 4s_v$ , where  $s_v$  is the standard deviation of the corresponding variable (cart or pole velocity) in the data collected along the whole set of studies with human users in this research<sup>5</sup>. The concrete values are  $\pm 6.304$  metres per second and  $\pm 4.510$  radians per second for the cart and pole velocities, respectively<sup>6</sup>.

The preconditions of a rule  $r$  define a “box” in the space of possible states of the pole and

<sup>5</sup>Data from humans (Section 5.3) was preferred because it represents a broader variety of strategies (and consequently, a broader sample of the set of possible states of the device) than the single strategy implemented in the artificial controllers. The specific threshold of  $\pm 4s_v$  is somewhat arbitrary, yet it guarantees only very rare situations are not covered.

<sup>6</sup>Compare this with the ranges defined by the corresponding thresholds estimated from the performance of the artificial controllers:  $\pm 4.280$  and  $\pm 4.837$ , respectively.

cart. The “volume” of this box can be defined as

$$V(r) = \prod_{v \in \{x, \dot{x}, a, \dot{a}\}} (\max(v) - \min(v)) \quad (5.5)$$

where  $\min(v)$  and  $\max(v)$  denote the minimum and maximum values of variable  $v$  that satisfies the condition of rule  $r$ , respectively. For preconditions where either the minimum or maximum value of a variable is not specified, the corresponding absolute minimum or maximum values is used instead—see Section 4.4.3 for a description of the general format of the preconditions of the rules in the models.

Intuitively, two rules  $r_a$  and  $r_b$  are similar if they both apply to similar states of the pole and cart and produce the same action; that is to say, if the boxes defined by their respective preconditions intersect to a great extent—and have the same “colour”. If we consider the subsets defined by the union and the intersection of the boxes in which  $r_a$  and  $r_b$  apply, and denote their volume by  $V(r_a \vee r_b)$  and  $V(r_a \wedge r_b)$ , respectively, then the measure of similarity between two rules  $r_a$  and  $r_b$  is defined as

$$S(r_a, r_b) = \left\{ \begin{array}{ll} +1 & \text{if both rules have the same action} \\ -1 & \text{if the rules have different pushing action} \\ 0 & \text{otherwise} \end{array} \right\} \times \frac{V(r_a \wedge r_b)}{V(r_a \vee r_b)}. \quad (5.6)$$

The measure of similarity between two models  $M_a$  and  $M_b$  is defined using the formula

$$S(M_a, M_b) = \frac{1}{\max(\#M_a, \#M_b)} \sum_{(r_a, r_b) \in R_G} S(r_a, r_b), \quad (5.7)$$

where  $\#M_a$  and  $\#M_b$  are the number of rules in  $M_a$  and  $M_b$ , respectively, and  $R_G$  is the relation between rules in  $M_a$  and  $M_b$  that results from matching rules in  $M_a$  to rules in  $M_b$ , until all the rules in the smaller model have been matched. The relation  $R_G$  is constructed iteratively using a *greedy* algorithm that each time chooses the pair  $(r_a, r_b)$  with the similarity value of greater magnitude, irrespectively of its sign<sup>7</sup>. Consequently, the definition of similarity between models penalises for highly similar rules with opposite actions.

Table 5.3 shows the degree of similarity between the models of the artificial controllers and the original set of rules. Models of  $\mathcal{V}_{250,x}$  achieved the highest similarity to the original set

<sup>7</sup>If more than one pair with maximum absolute value exist—say  $p_1 = (r_a^i, r_b^j)$  and  $p_2 = (r_a^k, r_b^l)$ , where  $i, j, k$  and  $l$  stand for the ranks of the rules in their respective models—the pair with the smallest average ranking is selected—e.g.  $p_1$  if  $(i+j)/2 < (k+l)/2$ . If more than one pair still exist, the pair containing the rule with the smallest ranking is selected—e.g.  $p_1$  if  $(i+j)/2 = (k+l)/2$  but, say,  $j < k$  and  $j < l$ . If even then more than one pair exist, then one containing the rule  $r_a$  with the smallest ranking is selected—e.g.  $p_2$  if  $\min(i, j, k, l) = j = k$  and hence  $k < i$ . The last decision makes the measure of similarity asymmetrical; it is included to complete the definition of the algorithm, though it was never needed in this research.

**Table 5.3:** Summary information about the contents of the models of the artificial controllers and their degree of similarity to the original set of rules (Table 5.1).

Controller	Initial state	Models		Similarity measure
		<i>Push-left</i> rules	<i>Push-right</i> rules	
$\mathcal{F}_{300,0}$	Easy centred	3	3	0.69
$\mathcal{F}_{300,10}$		3	4	0.68
$\mathcal{V}'_{250,0}$		4	3	0.71
$\mathcal{V}'_{250,10}$		3	3	0.69
$\mathcal{V}'_{350,0}$		2	2	0.19
$\mathcal{V}'_{350,10}$		1	1	0.21
$\mathcal{F}_{300,0}$	Semi random	3	3	0.70
$\mathcal{F}_{300,10}$		6	4	0.62
$\mathcal{V}'_{250,0}$		3	3	0.70
$\mathcal{V}'_{250,10}$		4	3	0.70
$\mathcal{V}'_{350,0}$		3	4	0.51
$\mathcal{V}'_{350,10}$		3	3	0.33
$\mathcal{F}_{300,0}$	Hard centred	3	3	0.36
$\mathcal{F}_{300,10}$		3	3	0.52
$\mathcal{V}'_{250,0}$		4	5	0.83
$\mathcal{V}'_{250,10}$		4	3	0.55
$\mathcal{V}'_{350,0}$		3	3	0.27
$\mathcal{V}'_{350,10}$		2	6	0.26

of rules, whereas models of  $\mathcal{V}'_{350,x}$  are the least similar to it. Statistically, analysis of variance shows that reaction delay is the only factor with a significant effect on model similarity to the real strategy (interaction effects are non-significant). A post-hoc Tukey-HSD test shows the only significant difference is between models of  $\mathcal{V}'_{350,x}$  and the rest; i.e. there is no significant difference between models of  $\mathcal{V}'_{250,x}$  and models of  $\mathcal{F}_{300,x}$  (Table 5.4).

Table 5.5 shows the model of  $\mathcal{V}'_{250,0}$  with the best match to the original strategy. The correspondence between rules in the model and rules in the original set (Table 5.1) is presented

**Table 5.4:** Summary of one-way analysis of variance of similarity by action delay.

One-way analysis of variance					
Similarity measure by action delay					
Source	D.F.	Sum of squares	Mean squares	F ratio	<i>p</i>
Between Groups	2	.5209	.2604	20.0223	.0001
Within Groups	15	.1951	.0130		
Total	17	.7160			

Multiple Range Tests: Tukey-HSD test with significance level .05

A tick (✓) indicates significant differences.

Mean	$\mathcal{F}_{300,x}$	$\mathcal{V}_{250,x}$	$\mathcal{V}_{350,x}$
.5940	$\mathcal{F}_{300,x}$		✓
.6964	$\mathcal{V}_{250,x}$		✓
.2954	$\mathcal{V}_{350,x}$		

in the first column using the notation  $n/m$ , where  $n$  is the rule number in the model and  $m$  is the number of the corresponding rule in the original set of rules. Many of the rules in the model of  $\mathcal{V}_{250,0}$  (rules 3, 5, 6, 7, 8, 9 and 10) resemble closely the original rules<sup>8</sup>; others, like rules 1 and 4, differ solely in some of the threshold values. Only rule 2 and its associated rule in the original rule set are completely different, their actions included; they are put together simply because they are the only rules that remain to be matched.

Table 5.6 presents a model of average similarity to the original strategy—one of  $\mathcal{F}_{300,10}$ . In this case, only two good matches have been achieved (rule 2, and the default rule); rules 1, 4, 5 and 6 have the same structure as their matching ones, differing only in some threshold values. The difference between the threshold values of rule 3 in the model and rule 4 in the original set is particularly noticeable.

Finally, Table 5.7 shows the model of  $\mathcal{V}_{350,0}$ , which achieved the lowest similarity measure to the original strategy. In this case, although some rules have a very good match to rules in the original set (rule 3 and the default rule), the first two rules are matched to rules with opposite actions.

<sup>8</sup>The structure of the models and the design of the matching algorithm guarantee that the default rule in every model be associated with the default rule in the original rule set.

**Table 5.5:** Model with the best match to the original set of rules (model of  $\mathcal{V}_{250,0}^2$ ).

1/6	left	104	9	if $x \leq -1.034$ and $\dot{x} \leq 0.373$ and $a \leq 0.058$ and $\dot{a} \leq 0.466$ .
2/1	right	647	84	if $a \geq 0.065$ and $\dot{a} \geq -0.268$ .
3/8	left	699	96	if $a \leq -0.071$ and $\dot{a} \leq 0.461$ .
4/2	right	84	10	if $x \geq 1.151$ and $\dot{x} \geq -0.177$ and $a \geq -0.064$ and $\dot{a} \geq -0.549$ .
5/4	right	718	100	if $a \geq 0.073$ and $\dot{a} \geq -0.548$ .
6/3	right	419	70	if $\dot{x} \geq 0.361$ and $a \geq -0.058$ and $\dot{a} \geq -0.478$ .
7/9	left	449	81	if $\dot{a} \leq -0.514$ .
8/5	right	449	81	if $\dot{a} \geq 0.514$ .
9/7	left	436	89	if $\dot{x} \leq -0.361$ and $a \leq 0.065$ and $\dot{a} \leq 0.549$ .
10/10	wait	15	17	.
use_best_rule				

**Table 5.6:** A model with average similarity to the original set of rules (model of  $\mathcal{F}_{300,10}$ ).

1/8	left	622	73	if $a \leq -0.073$ and $\dot{a} \leq 0.096$ .
2/9	left	367	46	if $\dot{a} \leq -0.510$ .
3/4	right	345	44	if $a \geq -0.077$ and $\dot{a} \geq 0.536$ .
4/7	left	566	76	if $\dot{x} \leq 0.443$ and $a \leq 0.072$ and $\dot{a} \leq 0.495$ .
5/5	right	703	96	if $a \geq 0.073$ .
6/2	right	148	28	if $x \geq 1.338$ and $a \geq -0.063$ and $\dot{a} \geq -0.495$ .
7/10	wait	15	7	.
use_best_rule				

**Table 5.7:** The model with the worst match to the programmed set of rules (model of  $\mathcal{V}_{350,0}^2$ ).

1/5	left	641	375	if $a \leq -0.069$ .
2/9	right	642	376	if $a \geq 0.069$ .
3/4	right	332	252	if $a \geq -0.070$ and $\dot{a} \geq 0.503$ .
4/8	left	329	251	if $a \leq 0.068$ and $\dot{a} \leq -0.506$ .
5/10	wait	53	82	.
use_best_rule				

### 5.2.4 Discussion

There are many ways to define a set of rules that produce the same behaviour as the set of rules programmed in the artificial controllers. The definition given in Table 5.1 was chosen with knowledge of the shape of the learner models produced by the method explained in the previous chapter; other definitions would have produced completely different results. The point of comparing the content of the models to the original rules is to show that the learner modelling method can recover a set of rules in the presence of noise. The noise can be the result of varying reaction time or simple slips.

The results suggest that the models with the highest degree of similarity reproduce the shape of the preconditions of the original rules and certain amount of details. Model with an average match still reproduce a big part of the shape of the original preconditions, but the threshold can be very different.

The degree of similarity responds to changes in reaction delay (Table 5.4), which in turn affects the amount of noise in the training data (Figure 5.2). In general, similar and off-line predicting power correlated well ( $r = 0.72$ ,  $F = 18.067$ ,  $p < 0.0006$ ), although the model of  $\mathcal{F}_{300,0}$  under hard-displaced conditions was an exception to the rule. What happened in this particular case was that the controller never recovered from the difficult initial condition, and hence exhibited a very limited behaviour. The model fits to the behaviour and it is able to predict it well, but fails to take into account more varied situations.

The combined results support the claim that the models of the artificial controllers reproduced in many cases the set of rules programmed in the controllers and in most of the cases were as good at predicting off-line as the very set of rules programmed in the controllers. On-line prediction was more difficult because noise product of reaction delay could not be removed in advance.

## 5.3 Evaluation of models of novice human controllers

Besides the previous study on modelling artificial controllers, three more studies and one experiment constitute the empirical work of this research<sup>9</sup>, all of them involving novice human controllers of the pole and cart:

---

<sup>9</sup>The following distinction is made: an *experiment* tests a *hypothesis* of the effect on a *dependent variable* of changes in an *independent variable* (which is, to some extent, controlled by the experimenter); a *study* measures a dependent variable but either does not have a hypothesis to test or the researcher does not have control over the independent variable (Coolican, 1994, Part II).

**Study 1** was designed as a pilot test of the method for learner modelling. Six people took part in the study, voluntarily and without payment (five of them were graduate students in Artificial Intelligence). A brief verbal explanation of the interface to the simulator of the pole and cart was given to each participant, followed by a request to try keeping the pole balanced (i.e. non-horizontal) and the cart inside the window. After five minutes of control, the participants were requested to continue for another five minutes, while prompted to try harder in pursuing the goal. All control runs started with the pole and cart in easy-centred state (Section 3.3), and the participants were encouraged to start a new control run after every crash.

A more detailed account of this study and its results (obtained using an early version of the modelling method) can be found in (Morales & Pain, 1999).

**Study 2** was designed and conducted for testing some improvements introduced in the modelling strategy after the analysis of the results of Study 1. Nine people participated in this study: five of them were the same as in Study 1 (to observe how the enhanced method worked on them) and the rest were other graduate students in Artificial Intelligence. The experimental conditions were the same as in Study 1, except for the fact that semi-random states were used instead of easy-centred ones to start control runs.

**Study 3** was designed for testing the comprehensibility of the graphical interface to the learner models. Thirteen people took part in it, and it is fully described in Chapter 6.

**Experiment 1** was carried out for testing the effects of participative learner modelling on the posterior behaviour of learners. It consisted of two sections, labelled **Experiment 1a** and **Experiment 1b**, with fifteen participants in each section. They are fully described in Chapter 7.

The participants in all the studies were requested to control the pole and cart for at least seven minutes without any other task to distract them. In Experiment 1, the participants have a familiarisation period of one minute and five minutes of control before being involved in other tasks. Consequently, the learner models analysed here were constructed from data that excluded a first minute of practice and included the following five minutes of control (four minutes for the participants in Experiment 1). The models were tested using data from the following minute of control.

### 5.3.1 Off-line prediction

Figure 5.4 presents the error rates at off-line predicting by the learner models of the participants in the studies and experiment summarised above (henceforth referred to collectively as ‘studies’). It includes also the error rates of RIPPER’s original output and of guessing in favour of the most frequent action. An overall comparison (paired  $t$ -test) between the error rates of the learner models and the error rates of RIPPER’s original output shows there is not significant difference between them. On the other hand, the error rates of the models greatly differ from the error rates of guessing in favour of the most frequent action: a mean error rate of 29% compared with a mean of 52.8% ( $t = 17.58$ ,  $df = 57$ ,  $p < 0.0001$ ; Wilcoxon’s  $T = 9$ ,  $z = 6.554$ ,  $p < 0.0001$ ). There are five cases, however, in which the error rates fall within each other’s interval of 95% confidence (two in Study 1, two in Study 2, and one in Experiment 1a). Analysis of variance (unbalanced data) shows no significant difference in the error rates of the learner models between experimental settings.

### 5.3.2 On-line prediction

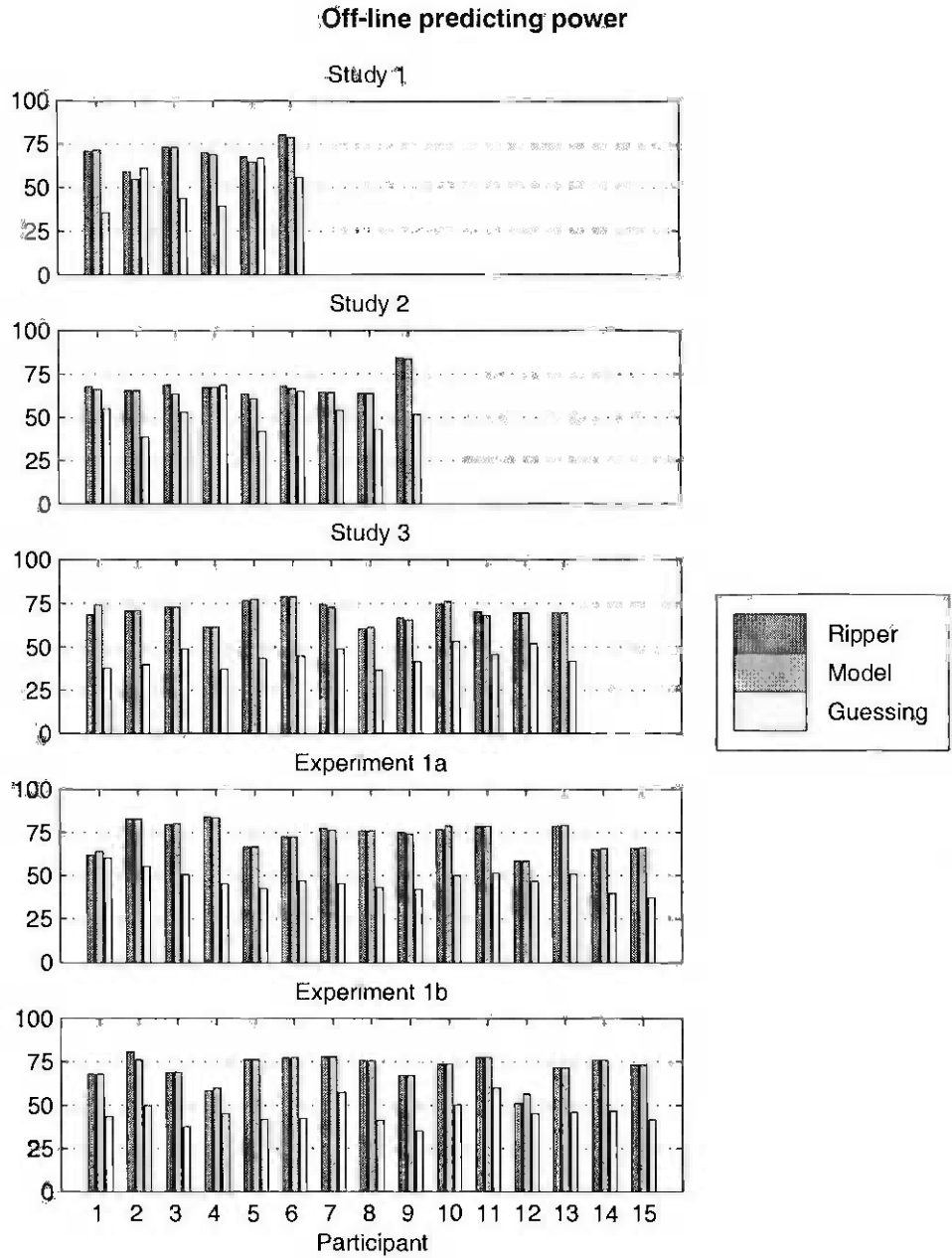
Figure 5.5 shows the error rates at predicting on-line by the learner models of the participants in the studies, together with the error rate of guessing in favour of the most frequent action (*wait* in most cases, except for Participant 2 in Experiment 1b). The first interesting fact to notice is that on-line prediction with the models is overall better than guessing, contrary to what happens with the models of the artificial controllers. The difference in error rates is not big though: a mean error rate of 30.6% compared with a mean of 38.3% ( $t = 3.155$ ,  $df = 57$ ,  $p < 0.003$ ; Wilcoxon’s  $T = 501$ ,  $z = 2.745$ ,  $p < 0.007$ ). The exceptions to the rule are the models constructed in Study 1, all of which are less powerful than simple guessing.

### 5.3.3 Contents of the models

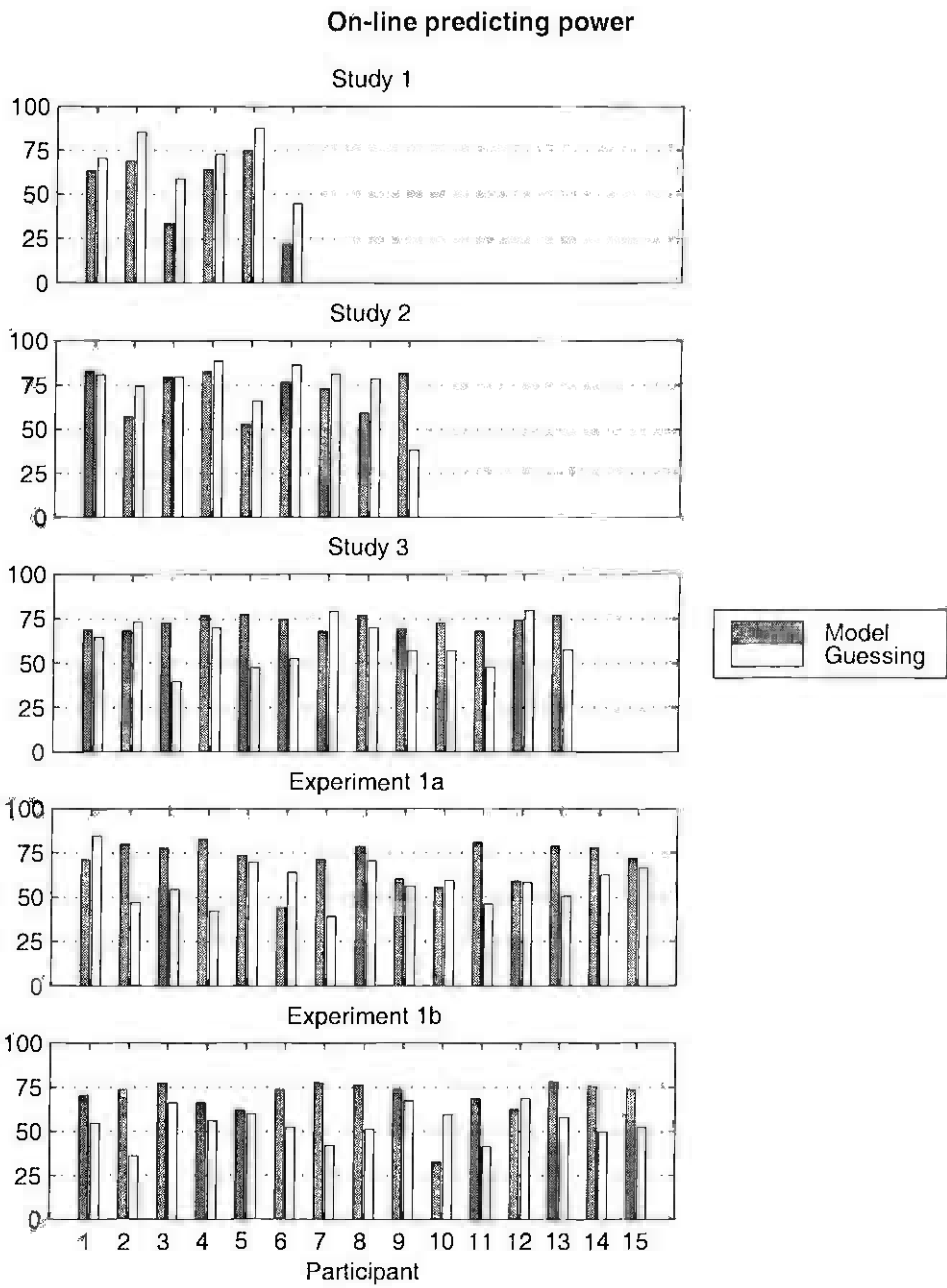
Overall, the learner model of the participants in the studies have between 5 and 17 rules ( $\bar{x} = 11.0$ ,  $s = 2.6$ ). The number of conditions per rule per model varies from 1.6 to 3.5, with an overall average of 2.6 conditions per rule.

The difference between the number of rules for *push-left* and *push-right* actions in a model is a rough estimate of the symmetry of the model: the smaller the difference, the greater the symmetry. The histogram in Figure 5.6 illustrates the distribution of frequencies of this estimator among the models of the participants in the studies. Although only 6 models have the same number of rules for *push-left* and *push-right* actions, the difference in the number of rules for

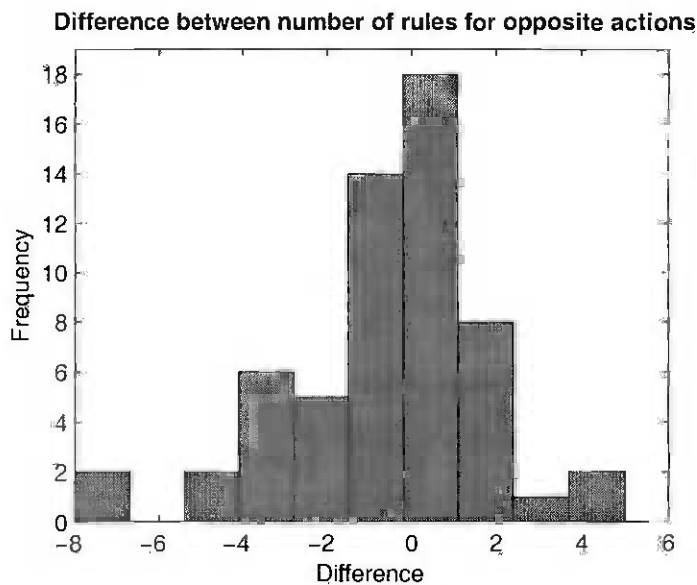




**Figure 5.4:** Off-line predicting power of the models of all participants in the studies. The predicting power of the corresponding original set of rules produced by RIPPER, and of guessing in favour of the most frequent action are included for comparison.



**Figure 5.5:** On-line predicting power of the models of all participants in the studies, compared to the predicting power of guessing in favour of the most frequent action.



**Figure 5.6:** Histogram of the difference per model between the number of rules for *push-left* and *push-right* actions.

each action is at most one in 32 models.

5.3.4 Discussion

The models of the novice human controllers have error rates at off-line predicting that lie in between the error rates of the models of the artificial controllers  $\mathcal{V}_{250,x}$  and  $\mathcal{V}_{350,x}$ , suggesting that novice human controllers were more inconsistent in their behaviour than  $\mathcal{F}_{300,x}$ . Unless the slip rate of the novice controllers were much higher than 10% of total actions, their inconsistency can be attributed either to variations in their reaction time or variations in their strategy. Explanations of variations in reaction time can be that the novice controllers reacted slower than their artificial counterparts; that they were able to foresee trends of behaviour and prepared their actions, hence reducing reaction time; and that they adapted to the task, becoming faster at perceiving important changes, selecting the appropriate action and executing it. Variations in strategy can be explained in terms of learning.

The models of the novice controllers are generally better at predicting on-line than the models of the artificial controllers. The explanation of this fact is that the models of the novices predict more *wait* actions in average than the models of the artificial controllers (33% and 9% of all their predictions, respectively). Since *wait* was the most frequent action “executed” by both the artificial and human controllers, the models of the latter perform better when the data

is not corrected for reaction time (Section 4.4.2).

Finally, there seems to be a weak correlation between symmetry and off-line predicting power ( $r = 0.17$ ,  $F = 0.69$ ,  $p < 0.17$ ). It might be the case that the difference in the number of rules with opposite actions is too rough an estimator of symmetry to make the relationship stronger. A more accurate estimator must take into account the preconditions of the rules.

## 5.4 Conclusions

The evaluation of the learner models of the participants in the studies carried out in this research suggest that they can be considered predictive of learner behaviour. An average predicting power of over 70% is a good result, although not excellent. The best justification of that may be that novice controllers of the pole and cart, who have only had one minute of familiarisation with the task, are simply too inconsistent to be modelled accurately with the method described in Chapter 4. But, if that were the case, a 70% accuracy at predicting learner behaviour should be considered a very good outcome, given the relative simplicity of the modelling method and the models that result from it. It is possible that including more detailed information about the learner (e.g. an estimation of her time to react to different situations) and the task (e.g. critical vs non-critical situations) would produce more powerful models.

To consider the task of inspecting and modifying the current learner models, it was necessary to produce evidence that they are, effectively, models of the learner. The results presented in this chapter suggest this requirement has been fulfilled to the extent that the models are predictive of learner behaviour. It remains to be shown that they can be presented to the learners in an informative way. This is considered in the next chapter.

## Chapter 6

# Presentation of learner models

El ciervo va a beber y en el agua aparece  
el reflejo de un tigre.  
El ciervo bebe el agua y la imagen. Se vuelve  
antes que lo devoren (cómplice, fascinado)  
igual a su enemigo.

Rosario Castellanos (1972). *Destino*.

This chapter describes how PACMOD presents the learner models to the learners and the facilities it provides for the learners inspecting and editing their learner models. It discusses the issues of modularity of model content and modality of its presentation, the latter specially in relation to the modality used for presenting the domain task (controlling the pole and cart). Finally, it reports two studies conducted for evaluating the interface to the learner models, in terms of how well the learners can understand their models.

### 6.1 From learner models to understandable learner models

Models constructed as explained in Chapter 4 provide useful information for a number of purposes: given a state of the pole and cart, a model predicts whether its learner will push to the left, to the right, or will do nothing; executed by a simple interpreter, a model can simulate its learner's behaviour; statistical comparisons between predictions of distinct models can indicate (dis)similarity between the strategies of their learners<sup>1</sup>.

---

<sup>1</sup>Learner models as predictors of learner behaviour were discussed in detail in Chapter 5, where statistical comparisons between their predictions were used to evaluate the method of learner modelling described in Chapter 4. An example of how learner models can be used as simulators of learner behaviour, as well as other uses of statistical comparisons between model predictions, can be found in (Morales & Pain, 1999).

For tutoring purposes, comparing a model with a set of rules representative of expertise can lead to situations (combinations of positions and velocities) in which the reaction of the learner represented by the model is different to the reaction of an expert. A common feature among all these different uses of learner models is that they do not require the learner models to be comprehensible to their learners. In fact, they hardly need human intervention at all: all that is needed is that a computer system can interpret the models. The situation is different when it comes to human understanding and manipulation of the models, particularly when these humans are precisely the learners represented by the models.

Several factors come into play that make learner models such as the one shown in Figure 4.4, presented again in Figure 6.1(a), hard to comprehend by learners. Some factors pertain to the content of the models; others relate to the way the models are presented; yet others have to do with how learners approach their models. Some of these factors can be tackled successfully by extending in simple ways the final stage of the modelling process described in Section 4.4.4; others may be addressed, also with good results, by designing a suitable user interface; others, however, are very hard to come to grips with.

The remainder of this section describes three simple manoeuvres to facilitate learners' understanding of their models: an alternative presentation of the models (Section 6.1.1), further refinements of their content (Section 6.1.2) and the provision of interactive facilities for learners to further explore their models (Section 6.1.3). Some major obstacles to learner understanding not solved by these simple steps are discussed in Section 6.4.

### 6.1.1 Presenting the models in the right modality

Rule preconditions are expressed in numerical terms in the model shown in Figure 6.1(a). Although they can be very accurate, they are quite difficult to correlate to the actual behaviour of the simulated pole and cart on the screen. A different choice of units (e.g. degrees instead of radians, or millimetres instead of metres) does not alleviate the problem, because it is not a problem of units but one of *modality*<sup>2</sup>: the lack of consistency in the way the task (Figure 3.2) and the learner models (e.g. Figure 6.1(a)) are presented makes the latter more difficult to comprehend, despite experience with the former. For example, it is very difficult to realise which states of the pole and cart satisfy the preconditions of the fifth rule in the example model,

---

<sup>2</sup>The term 'modality' is used here in an intuitive and informal way to imply a combination of medium and language, with no attempt to give a more precise definition of it. For a more in depth discussion of modality see (Pineda & Garza, 1999).

left 383 83 if  $x \geq -0.794$  and  $a \leq -0.073$  and  $\dot{a} \leq -0.022$ .





without some careful analysis. Even after the analysis has shown that the pole has to be tilted to the left and rotating anticlockwise not too slowly, it is still hard to *see* how tilted the pole has to be, and how slow is slowly.

A different way of presenting learner models is shown in Figure 6.1(b). The model shown originally in Figure 6.1(a) is presented here graphically, in a table-like format in which every row represents a rule. The first column in the presentation, headed No., contains buttons numbering the rules—other functions of these buttons are described in Section 6.1.3. The last column, headed ACTION, may contain the arrows  $\leftarrow$  and  $\rightarrow$ , and the word WAIT, which represent the actions *push-left*, *push-right* and *wait*, respectively. The remaining columns, from second to fifth, contain graphical representations of rule preconditions:

- the second column, headed POLE ANGLE IN, contains drawings of arcs, each one denoting a range of angles for the pole;
- the third column, headed POLE VELOCITY BETWEEN, contains animations of the pole moving with different velocities—every pair of animations denotes the range of velocities between the velocities of the animations;
- the fourth column, headed CART POSITION IN, contains drawings of boxes that represent ranges of positions for the cart; and
- the fifth column, headed CART VELOCITY BETWEEN, contains animations of the cart moving with different velocities—as before, every pair of animations denotes the range of velocities between the velocities of the animations.

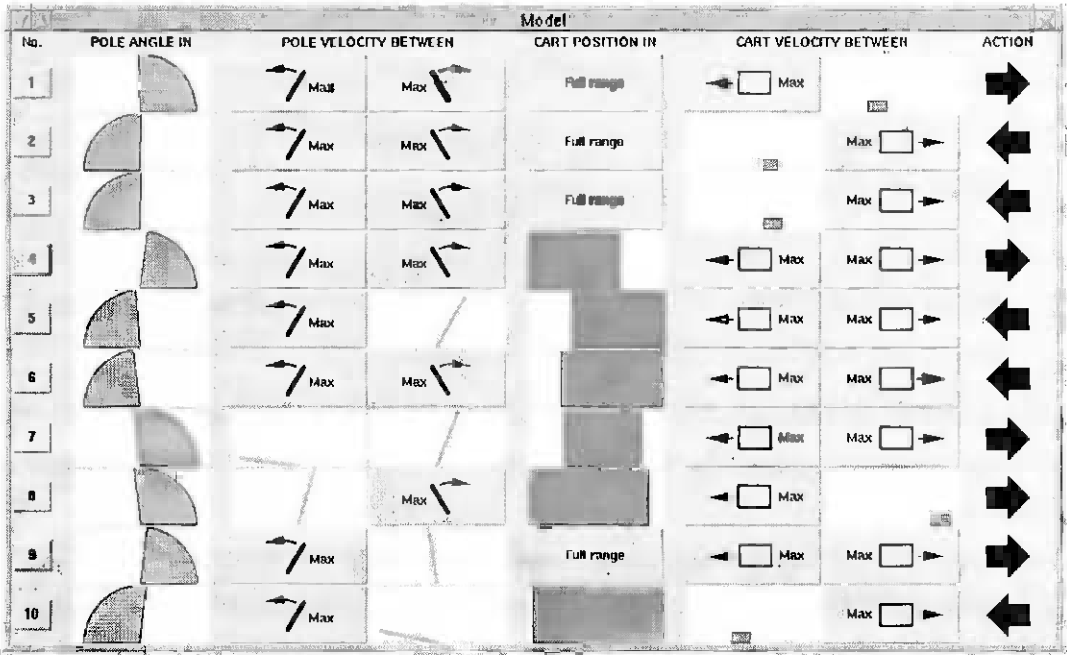
Velocities consist of both magnitude and direction; the convention used is that they go from very fast anticlockwise to very fast clockwise (pole velocities), and from very fast to the left to very fast to the right (cart velocities).

The graphical presentation of a learner model may contain also the phrase FULL RANGE under POLE ANGLE IN and CART POSITION IN to denote the full range of angles and positions, respectively. It can also contain the following icons:

-  for maximum speed anticlockwise,
-  for maximum speed clockwise,
-  for maximum speed to the left, and
-  for maximum speed to the right.

right 163 20 if  $\dot{x} \leq -1.454$  and  $a \geq -0.029$  .  
left 164 24 if  $\dot{x} \geq 1.446$  and  $a \leq 0.030$  .  
left 163 24 if  $\dot{x} \geq 1.280$  and  $a \leq 0.017$  .  
right 383 76 if  $x \leq 0.852$  and  $a \geq 0.145$  .  
left 383 83 if  $x \geq -0.794$  and  $a \leq -0.073$  and  $\dot{a} \leq -0.022$  .  
left 440 110 if  $x \geq -1.214$  and  $a \leq -0.132$  .  
right 159 43 if  $-1.086 \leq x \leq 1.592$  and  $a \geq -0.098$  and  $0.330 \leq \dot{a} \leq 0.973$  .  
right 328 92 if  $x \leq 1.931$  and  $\dot{x} \leq 1.079$  and  $a \geq -0.100$  and  $\dot{a} \geq 0.274$  .  
right 378 121 if  $a \geq 0.120$  and  $\dot{a} \leq 0.892$  .  
left 325 110 if  $x \geq -2.163$  and  $\dot{x} \geq -1.426$  and  $a \leq 0.118$  and  $\dot{a} \leq -0.348$  .  
wait 0 0  
use\_best\_rule

(a) Learner model



(b) Graphical presentation

**Figure 6.1:** Snapshot of the graphical presentation of a learner model in PACMOD. The same model was shown in its symbolic form in Figure 6.1(a). The default rule is omitted because it is assumed always to produce a *wait* action..



Phrase and icons are introduced to improve the clarity of the presentation and to diminish the overwhelming effect of seeing too many fast animations.

The domain task and the learner model are presented now in the same graphical way. Consequently, much less translation from the notation used to present the model to the memories of pole and cart behaviour should be necessary, and hence the learner's previous experiences with the task should better support his understanding of the model. For example, once the notation described above is understood, it should be easy to *see* in the graphical presentation of the rule used as an example before,



which states of the pole and cart satisfy its preconditions. This is done without falling back on a wearisome analysis of the notation.

The consistency between the modalities in which the tasks in the domain and the learner models are presented have not been an issue raised explicitly in previous research on inspectable learner models. In cases of second language learning as the application domain, written natural language has been the modality of choice to present both the tasks and the models: the latter have been presented as grammatical rules with confidence measures attached (Bull & Pain, 1995; see also Section 2.2) as answers to previous exercises and statements describing general tendencies of learner behaviour (Bull & Broady, 1997); as qualitative evaluation of assignments, of overall performance, and free text comments (Bull, 1997b); and as statements of learners' communicative goals and capabilities (Ayala & Yano, 1996). In a similar way, de Buen et al. (1999) employ mathematical notation, specialised technical terminology and natural language to convey engineering concepts and procedural steps, to present tasks and learner model. In contrast, Cook & Kay (1994) present the models of users of the SAM text editor using a mixture of text and diagrams (conceptual trees) that differs from the traditional textual interface of the editor itself. Paiva et al. (1995) describe language selection as a major difficulty for presenting the learner models in a way that can be readily understood by learners, but the early example they provide shows the learner model in a language that is closer to the model's internal representation than to the language in which the task (simplification of algebraic equations) is presented.

The relevance of choosing similar ways to present the domain tasks and the learner models may vary from one domain to another. Sharing a modality may be crucial in domains like balancing the pole on the cart, where learning consists mainly on the acquisition of sensorimotor skills, because these skills depend heavily on quick recognition of patterns and automatic

reaction to them. However, sharing a modality may be less useful in domains that are more demanding of higher-level cognitive abilities, like the domains enumerated in the paragraph above. In these cases, it is probably less important that the presentations of the domain tasks and the learner models exhibit shallow similarities, and more important that they share structural properties or even a deeper but recognisable sets of abstractions. Furthermore, recent research has placed emphasis on the fact that different learners have different preferences of modality and different degrees of representational competence (Conlon, 1999; Conlon et al., 1999; Cox & Brna, 1995). These differences ought to be recognised in intelligent learning environments.

### 6.1.2 Increasing the modularity of the models

One problem with the learner model shown in Figures 6.1(a) and 6.1(b) is that the preconditions of its rules are not disjoint. In other words, the sets of states of the pole and cart that satisfy each rule's preconditions may overlap. For example, the preconditions of the second and seventh rules of the model,

left	164	24	if $\dot{x} \geq 1.446$ and $a \leq 0.030$ .
right	159	43	if $-1.086 \leq x \leq 1.592$ and $a \geq -0.098$ and $0.330 \leq \dot{a} \leq 0.973$ .

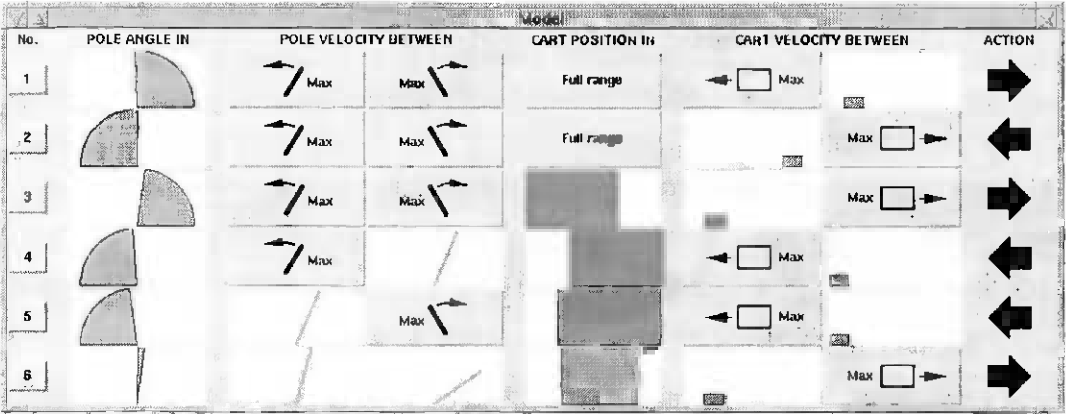
are matched by all the states of the pole and cart in which the angle of the pole is between  $-0.098$  and  $0.030$  radians ( $-5.61$  and  $1.72$  degrees), the pole is moving clockwise with a velocity in between  $0.330$  and  $0.973$  radians per second, the position of the cart is between  $-1.086$  and  $1.592$  metres, and it is moving off the window to the right faster than  $1.446$  metres per second—a better idea of what all this means can be obtained by inspecting the graphical presentation of the model in Figure 6.1(b). The decision of which rule is selected when a conflict such as this occurs is taken based on the order of the rules in the learner model.

The first row in Figure 6.1(b) contains the most important rule of the learner model, in the sense that the learner executes this rule's action at every state of the pole and cart that satisfies its preconditions. The second row contains the second most important rule; that is, its action is executed by the learner at every state of the pole and cart that satisfies this rule's preconditions but does not satisfy the preconditions of the rule in the topmost row. This continues in the same way until the rule presented at the bottom row, which it is selected by the learner only for states of the pole and cart that satisfy its preconditions but do not satisfy the preconditions of any previous rule— if no rule matches a state of the pole and cart, then it is assumed the learner normally does nothing (i.e. executes a *wait* action) in that situation.

Order dependency for rule firing has two potential benefits: it can reduce the number of rules in the model, and can simplify their preconditions. Nevertheless, it has also a undesirable side effect: the states of the pole and cart for which a rule is chosen are not determined completely by the rule's preconditions, but also by the preconditions of all rules with higher priority; as if every rule had the (negated) preconditions of all previous rules attached to it. This makes the individual role of every rule harder to comprehend, reduces their character as *independent pieces of knowledge* (Barr & Feigenbaum, 1981) and hence makes the learner model *less modular*.

right	2	18	163	if $\dot{x} \leq -1.454$ and $a \geq -0.029$ .
left	164	22	2	if $\dot{x} \geq 1.446$ and $a \leq 0.030$ .
right	13	63	330	if $x \leq 0.852$ and $\dot{x} \geq -1.454$ and $a \geq 0.145$ .
left	287	65	15	if $x \geq -0.794$ and $\dot{x} \leq 1.280$ and $a \leq -0.073$ and $\dot{a} \leq -0.022$ .
left	120	41	14	if $x \geq -1.214$ and $\dot{x} \leq 1.280$ and $a \leq -0.132$ and $\dot{a} \geq -0.022$ .
right	8	24	49	if $-1.086 \leq x \leq 1.592$ and $\dot{x} \geq -1.454$ and $0.030 \leq a \leq 0.145$ and $0.330 \leq \dot{a} \leq 0.973$ .

(a) Learner model



(b) Graphical presentation

**Figure 6.2:** Refined version of the learner model shown previously in Figures 6.1(a) and 6.1(b). In this, its final version, the preconditions of all rules are mutually exclusive, so that at most one rule matches any state of the pole and cart. The default rule is omitted because it is assumed always to produce a *wait* action.

Figure 6.2 contains a refined version of the model presented in Figure 6.1(b). This time the preconditions of all rules have been modified to make them mutually exclusive, with the beneficial side-effect, in this particular case, of a significant reduction in the number of rules—see Section 6.4 for overall figures of change in model size due to increased modularity. The new model is completely equivalent to the old one, to the extent that both models predict the same action for each possible state of the pole and cart, but now the role of each rule can be understood more independently of any other rule. The new model profits more than the old one from the long acclaimed modularity of production rules for representing knowledge (Barr & Feigenbaum, 1981; Hayes-Roth, 1985).

Modularity, as modality, has not been explicitly discussed in the context of previous research on inspectable models. Although distinct components that may favour a modular design can be observed in all inspectable models and their presentations—grammatical rules, description of general tendencies, answers to exercises and their evaluation, specific and more general comments, communicative goals and capabilities, knowledge components, user properties, beliefs and reasoning rules—their use does not intrinsically guarantee modularity, as our example using production rules clearly illustrates.

### 6.1.3 Providing Interactivity

The two steps described above for refining the contents and enhancing the presentation of learner models should give learners improved access to their models. Nevertheless, it cannot be guaranteed that learners will pay any attention to their models, nor that they will understand them. A way of boosting learners' interest in their models, without diminishing their initiative, is to supply them with means to interact with their models in more interesting ways than merely looking at them. Even more, it has been argued elsewhere that learners should feel they are able to exert some influence over the content and structure of their models, rather than merely observing them—e.g. Variation 15 on openness, 'Open = extended, expanded or unfolded', in (Self, 1999b); see also Chapter 1. Most researchers working on open learner modelling have encouraged learners to interact with their models; the facilities provided to that end varied from browsing and direct editing of (parts of) the learner models (Ayala & Yano, 1996; Bull, 1997b; Cook & Kay, 1994; de Buen et al., 1999; Paiva et al., 1995), to mechanisms for discussing and negotiating their content (Bull & Pain, 1995; Dimitrova et al., 1999a).

The interface to learner models that has been described here includes facilities for learners to explore their models and to edit the models' contents. The rule induction technique at the heart of the diagnosis process sorts the rules in a model in terms of the ratio of cases properly

classified by the rule to the total number of cases in which the rule applies (Sections 4.4.3 and 4.4.4). This information is stored in the learner models and is presented in the graphical interface implicitly in the order of the rules, but it is displayed explicitly on requested to justify the existence and relevance of every rule (Figure 6.3). This simple move should motivate learners to accept their models more readily.

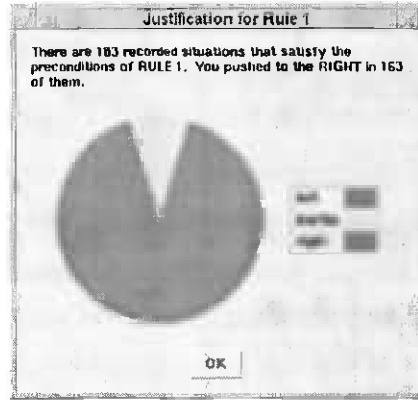


Figure 6.3: Justification of a rule.

The learners can ask PACMOD to execute the learner model starting with a state of the pole and cart that favours the execution of a particular rule. In response to such a request, PACMOD takes control of the pole and cart simulator and runs the learner model for a few seconds, highlighting the action of the rule being fired and signalling the same action explicitly in the control interface, using the same arrow notation as in the rules.

A further way of exploring the learner model is by requesting the explanation of any rule in textual form. The explanation consists of an *if...then* template filled with translations into words of the rule's preconditions and action (Figure 6.4). Unfortunately, this facility is in very rudimentary form: the range of each state variable (position, angle and velocities) is first split into subintervals<sup>3</sup> that are associated with the categories tiny, small, medium, big and huge; then each precondition is translated according to the subintervals in which its thresholds fall. Consequently, PACMOD produces explanations that are too verbose and repetitive, but they seem to be useful in clarifying the meaning of the graphical description of the rules (Section 6.3.3).

PACMOD gives learners direct control over the content of their models by enriching the presentation of the models with the means of editing the models (unless it is running in view-

<sup>3</sup>The limits of the subintervals were first reckoned empirically, based on the distribution of frequencies of the values of the state variables, and then adjusted by trial-and-error until they produced good results.

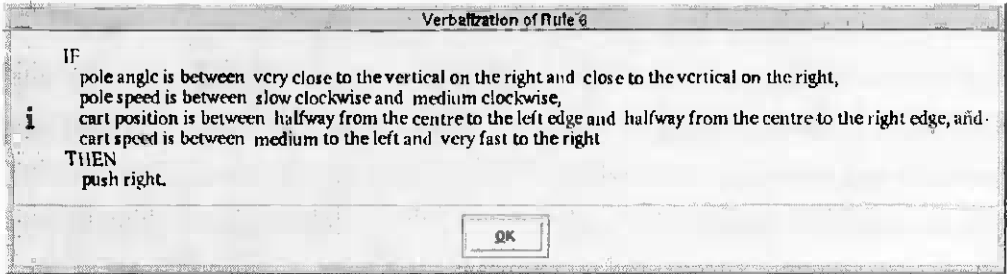


Figure 6.4: Verbalization of a rule.

only mode). Learners can modify the preconditions on pole angle and cart velocity of any rule by direct manipulation of their graphical presentation (this is done by dragging the borders of arcs and boxes, respectively); they can modify the preconditions on pole and cart velocity by dragging sliders on a graphical scale that appears when the mouse is over such a precondition; and they can change the action of any rule by clicking a button on the mouse. Learners can add, delete, and alter the order of rules in their models using the palette of operations shown in Figure 6.5—reordering is still important because disjoint preconditions among rules is not guaranteed after the learners have edited their models. The facilities for justification, verbalization and execution of rules, mentioned above, offer the possibility of immediate feedback to learners on modifications to their models.

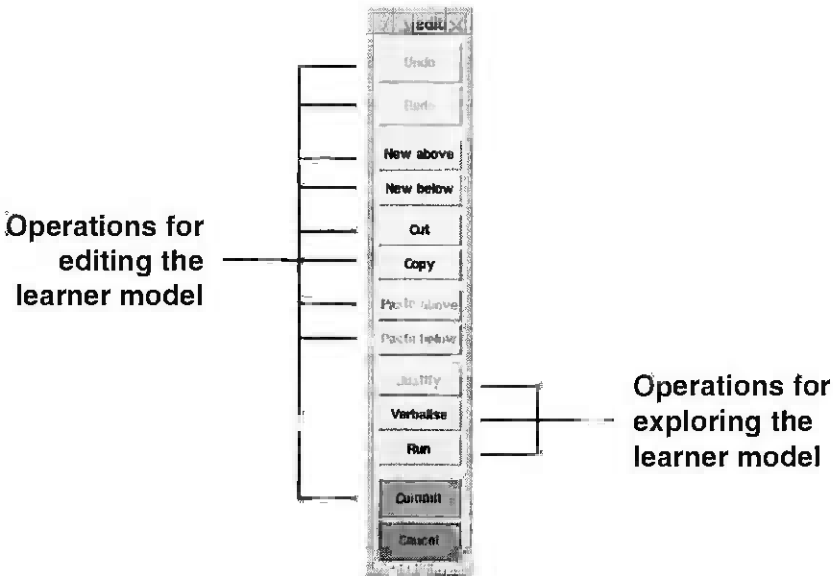


Figure 6.5: Palette of operations for exploring and modifying the learner model.

## 6.2 Evaluation of the interface I

The goal of Study 3 was to evaluate how easy it is to understand a set of rules when displayed using the graphical interface described above. The early version of PACMOD used in this study lacked the facility for verbalizing rules, presenting the short executions of rules as explanations of the target rule.

Eleven people took part in the study<sup>4</sup>: eight men and three women. Most of them were postgraduate students, in areas such as Artificial Intelligence (3), Computer Science and Engineering (3), Ecology (1), Agriculture (1) and Veterinary (1); the two remaining had first degrees in Arts and Literature, respectively. The participants' ages ranged from 26 to 41 years, with a mean and median of around 30 years.

### 6.2.1 Procedure

The study consisted of three stages. Firstly, the participants were requested to provide some background information by answering a small questionnaire—gender, age, most practised sports, taste for video-games and a self-assessment of proficiency at playing them. The second stage involved playing the pole and cart game for about nine minutes<sup>5</sup>. The third stage included reviewing the graphical presentation of a fixed set of rules, representing a fictitious strategy for controlling the pole on a cart (Figure 6.6), and answering a questionnaire about it. The questions were organised in (what it was believed to be) increasing order of difficulty, from simple questions that tested understanding of the arrow notation to represent actions to a final question aimed at testing understanding of the strategy as a whole. Four more questions about the general interestingness of the study were included at the end of the questionnaire.

The participants received instructions and questionnaires in printed form at the beginning of each stage (Appendix B), and further interventions of the experimenter were kept to a minimum; nevertheless, the participants were always allowed to ask for clarification of any aspect of the program and printed material. They could employ as much time as they wished in answering the questionnaires, so they spent between 35 and 75 minutes in the last stage of the experiment, mostly reviewing the set of rules and answering questions about it.

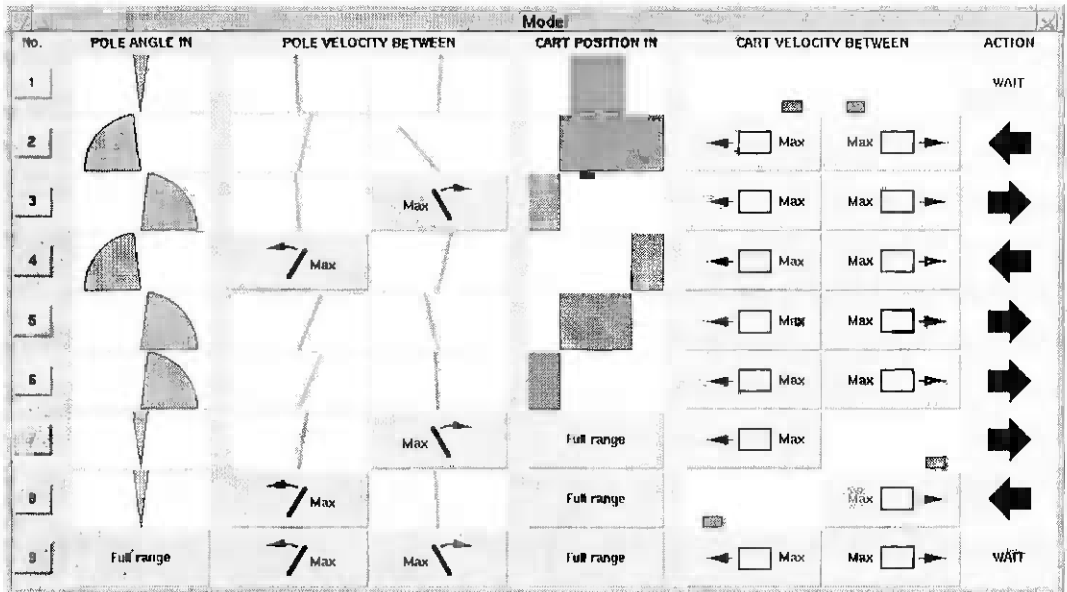
---

<sup>4</sup>Actually, they were thirteen, but data from two of them was discarded because the experimental conditions were unsuitable.

<sup>5</sup>Playing time varied due to a bug in the program. The average playing time was 8.4 minutes, with a minimum of 7.9 and a maximum of 9.7 minutes.

wait 0 8 0 if  $-0.825 \leq x \leq 0.975$  and  $-0.360 \leq \dot{x} \leq 0.360$   
 and  $-0.136 \leq a \leq 0.136$  and  $-0.160 \leq \dot{a} \leq 0.160$  .  
 left 7 0 0 if  $x \geq -1.275$  and  $a \leq -0.136$  and  $-2.240 \leq \dot{a} \leq 1.412$  .  
 right 0 0 6 if  $x \leq -1.275$  and  $a \geq 0.136$  and  $\dot{a} \geq -0.160$  .  
 left 5 0 0 if  $x \geq 1.275$  and  $a \leq -0.136$  and  $\dot{a} \leq -2.240$  .  
 right 0 0 4 if  $-1.275 \leq x \leq 1.275$  and  $a \geq 0.136$  and  $-1.178 \leq \dot{a} \leq 2.240$  .  
 right 0 0 4 if  $x \leq -1.275$  and  $a \geq 0.136$  and  $-1.178 \leq \dot{a} \leq -0.160$  .  
 right 0 0 3 if  $\dot{x} \leq 1.620$  and  $-0.136 \leq a \leq 0.136$  and  $\dot{a} \geq 0.320$  .  
 left 2 0 0 if  $\dot{x} \geq -1.620$  and  $-0.136 \leq a \leq 0.136$  and  $\dot{a} \leq -0.160$  .  
 wait 0 1 0 .

(a) Symbolic encoding of the set of rules



(b) Snapshot of the graphical presentation

Figure 6.6: A set of rules representing a fictitious strategy for controlling the pole and cart.



### 6.2.2 Results

The answers to the questionnaire were evaluated qualitatively: a model answer for each question was compared with answers from the participants. For example, the model answer to the question 'Explain in your own words the precondition on the pole velocity of rule 5' (Question 8) was

The pole is either falling to the left at moderate velocity, or it is falling to the right with an up to moderately fast velocity.

An answer to Question 8 that was taken as correct is

Pole velocity falling at moderate speed to left, but fast to right. (Participant 9)

An answer that was taken as partially (in)correct is

The pole is not falling left or right with maximum velocity. (Participant 8)

And an example of an incorrect answer to the same question is

The pole was in a small angle to the right and the cart was pushed to the left in any velocity. (Participant 4)

The model answer to the most difficult question, 'Try to provide a description, in your own words, of the overall strategy defined by the rule set' (Question 14), was

The strategy is a sort of "natural" one, with a touch of laziness. That means doing nothing if both pole and cart are centred and moving slowly, pushing in the direction of pole's falling if it accelerates, but not if the cart is moving fast in that direction, and in general trying to revert to a centred position as long as the cart is not too close to an edge (with the pole falling more than slowly towards the edge) or the pole is not falling too quickly.

An example of a correct answer to Question 14 is

If the pole is well balanced and the cart is moving slowly in the centre of the screen, do nothing. If the pole is pointing off to the left, is not falling rapidly to the right and the cart is not on the far left, push left. If the pole is pointing to the right and the cart is not on the far right push right.

If the pole is upright falling right and the cart is not travelling fast right, push right. Similarly if it is falling left, not going fast left, push left.

Otherwise do nothing. (Participant 8)

The outcome of the evaluation of the questionnaires is presented in Table 6.1. First of all, it suggests that the difficulty of the questions increased as expected; secondly, it shows that understanding of the set of rules varies among the participants from minimal (e.g. Participant 3) to very good (e.g. Participant 6). A measure of each participant’s understanding of the set of rules can be calculated from Table 6.1 by summing up all his correct answers; each answer weighed by the difficulty of the question<sup>6</sup> The difficulty of a question can also be estimated from Table 6.1 using the formula

$$\text{difficulty} = 1 - \frac{\# \text{ of correct answers} + 1}{\# \text{ participants} + 2} \tag{6.1}$$

which employs Laplace correction to estimate the probability of an incorrect answer (Good, 1965). Figure 6.7 shows the estimated difficulty of the questions and Figure 6.8 shows the scores corresponding to the level of understanding of the participants in the study.

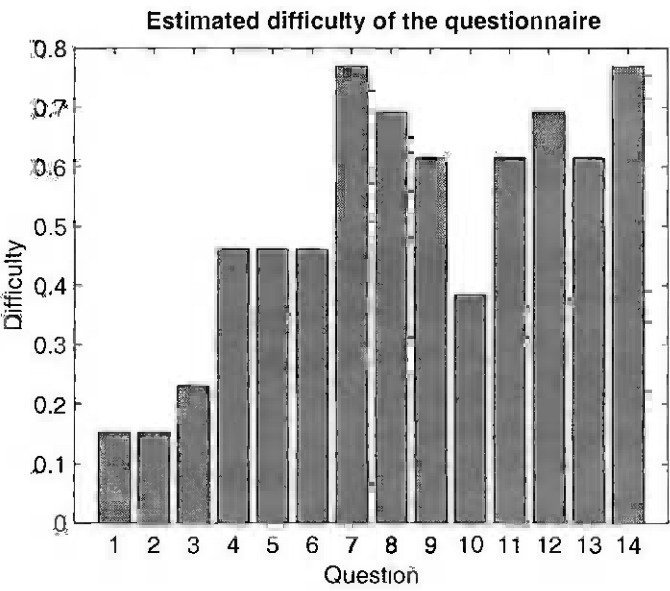


Figure 6.7: Estimated difficulty of the questionnaire.

6.2.3 Discussion

The results of the study, summarised in Table 6.1 and Figure 6.8, suggest considerable variability in understanding of the set of rules among the participants in the study. A result like this could be expected, to some extent, given the differences in background among the participants<sup>7</sup>.

<sup>6</sup>A measure which penalises for incorrect answers is used in (Morales et al., 2000). It (surprisingly) produces slightly better scores for the participants overall.

<sup>7</sup>The two lowest scores correspond to the participants with backgrounds in Art and Literature.

**Table 6.1:** Outcome of evaluation the answers to the main questionnaire of Study 3: ‘✓’ indicates a correct answer; ‘X’ represents an incorrect one; ‘?’ stands for a partially (in)correct answer; and ‘–’ indicates no answer.

Question	Participant										
	1	2	3	4	5	6	7	8	9	10	11
<i>On actions:</i>											
Q1	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
Q2	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>On cart position:</i>											
Q3	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Q4	?	X	X	✓	✓	✓	✓	?	?	✓	✓
<i>On pole angle:</i>											
Q5	✓	X	?	✓	✓	✓	✓	?	?	✓	X
<i>On pole velocity:</i>											
Q6	✓	X	–	✓	✓	✓	?	X	?	✓	✓
Q7	?	X	X	?	✓	?	X	?	?	X	✓
Q8	?	X	X	✓	✓	?	✓	?	?	?	?
<i>On cart velocity:</i>											
Q9	?	X	X	✓	✓	✓	?	?	?	X	✓
Q10	✓	X	?	✓	✓	✓	✓	✓	?	?	✓
<i>On whole rules:</i>											
Q11	?	X	X	✓	?	✓	?	?	X	✓	✓
Q12	?	?	X	✓	?	✓	?	?	?	?	✓
Q13	✓	?	X	✓	?	✓	?	?	?	X	✓
<i>On the strategy:</i>											
Q14	?	–	–	–	–	✓	–	✓	–	?	–

Assuming the scores shown in Figure 6.8 reflect levels of understanding in a linear way, with a score of zero representing no understanding and a maximum score representing total understanding, it can be said that in average the participants understood between 35% and 51% of what they saw on the screen—depending on whether partially (in)correct answers are taken into account, or not.

From the participants’ responses to the questionnaire, the doubts they expressed during

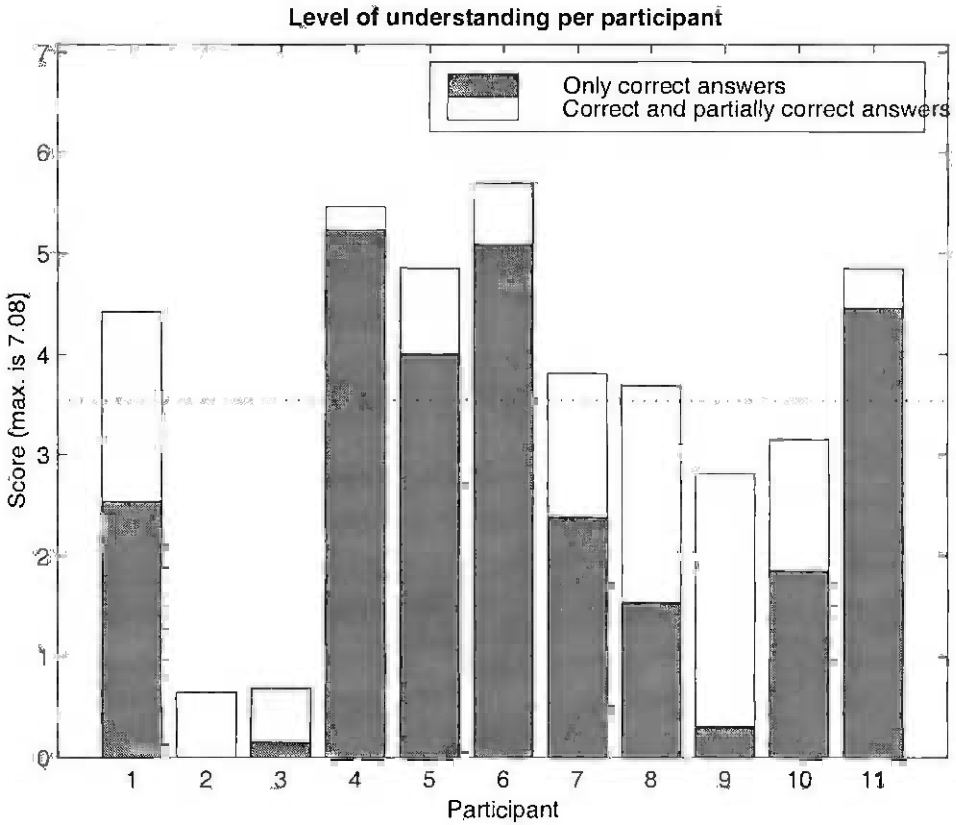


Figure 6.8: Level of understanding per participant.

the study and the relative difficulty of the questions shown in Figure 6.7, it became clear that the participants’ main problem was to interpret correctly the animations representing ranges of velocities. In addition, the participants frequently got confused by the “explanations” of the rules in terms of short executions of the strategy. Their confusion made evident their lack of confidence in their interpretation of the graphical interface, but also pointed out the need to enhance the interface in a number of ways.

Three means of facilitating a better understanding of the graphical presentation of learner models were suggested by the study, and are implemented now in PACMOD; they are:

1. Explanations of the rules in natural language, as a complement to the original “explanations by execution”.
2. Rewriting of the printed material, to make it clearer and more detailed.
3. Making explicit the action being executed at each moment in the short executions of the

learner model<sup>8</sup>.

Despite the small size of the study, and the fact that the analysis of the participants' responses to the questionnaire was carried out in an informal way, prone to subjective bias, the results obtained suggest that learners were able to achieve a reasonable understanding of their models.

## 6.3 Evaluation of the interface II

The purpose of Experiment 1 was to test the effects of participative learner modelling on learner behaviour, and it is described in detail in Chapter 7. Part of the experiment, however, had to do with using the graphical interface to learner models and answering questions about it; that is the part described here.

### 6.3.1 Procedure

The participants in the study were divided into three groups of 10 people each, and each group was allocated to one of three conditions to be tested: **Expert**, **Model** and **Practice** (Section 7.2). The members of the **Expert** group had to inspect a predefined set of rules for controlling the pole and cart (Figure 6.9), suggested to them as representing a good strategy to carry out the task. The members of the **Model** group, on the other hand, were presented with their learner models, limited to a maximum of twelve rules per model (see Appendix D). Finally, the members of the **Practice** group did not inspect any set of rules, but were given more practice in the domain task instead.

Five questions were posed to the participants to measure their understanding of what was presented to them. The first two questions were as in Study 3, to test whether the participants understood the arrow notation used to denote actions ('Which rules indicate an action of *pushing to the left*?'). The next three questions tested their understanding of the graphical notation used to display rule preconditions, and were slightly adapted to each condition. For example, Question 3 to the **Expert** group was

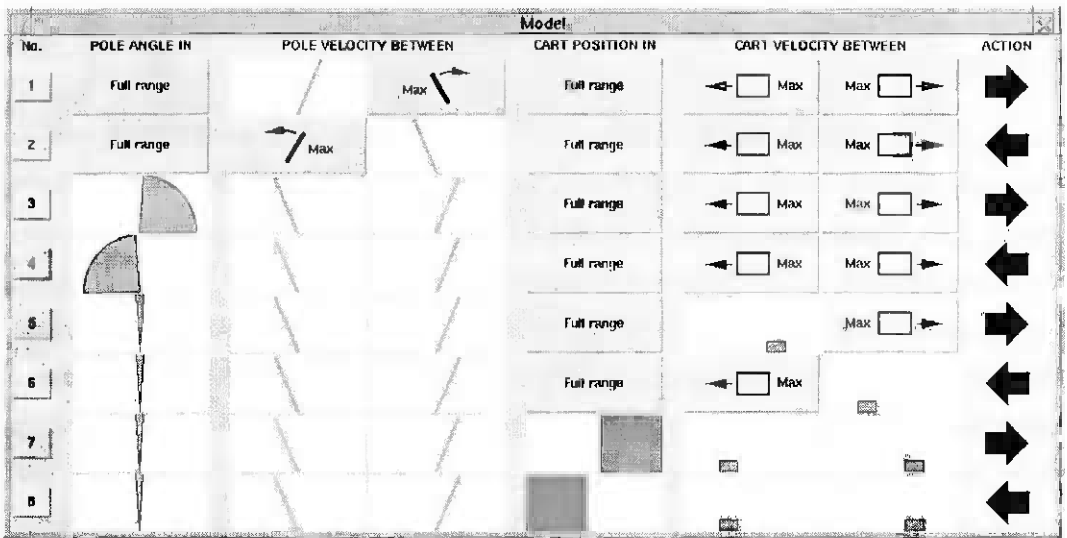
3. Which rule (if any) does the system believe the [skilled] user would select if the pole is halfway down and falling fairly quickly to its right, at the same time as the cart is halfway between the window's centre and left border and it is moving towards the left with moderate speed?

---

<sup>8</sup>The participants frequently took the direction in which the cart was moving as the action being executed.

right 0 0 10 if  $\dot{a} \geq 0.140$  .  
 left 9 0 0 if  $\dot{a} \leq -0.140$  .  
 right 0 0 8 if  $a \geq 0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .  
 left 7 0 0 if  $a \leq -0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .  
 right 0 0 6 if  $\dot{x} \geq 0.261$  and  $-0.068 \leq a \leq 0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .  
 left 5 0 0 if  $\dot{x} \leq -0.261$  and  $-0.068 \leq a \leq 0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .  
 right 0 0 4 if  $x \geq 0.289$  and  $-0.261 \leq \dot{x} \leq 0.261$   
 and  $-0.068 \leq a \leq 0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .  
 left 3 0 0 if  $x \leq -0.289$  and  $-0.261 \leq \dot{x} \leq 0.261$   
 and  $-0.068 \leq a \leq 0.068$  and  $-0.140 \leq \dot{a} \leq 0.140$  .

(a) Symbolic encoding of the rules



(b) Snapshot of the graphical presentation

Figure 6.9: Set of rules presented to the **Expert** group;

whereas the same question to the **Model** group was

- Which rule (if any) does the system believe you would select if the pole is halfway down and falling fairly quickly to its right, at the same time as the cart is halfway between the window's centre and left border and it is moving towards the left with moderate speed?

The last three questions were different in nature to the ones posed to the participants in Study 3: instead of asking the participants to explain in they own words what they saw on the screen, the new questions asked the participants to identify the rules that matched a qualitative description of a state of the pole and cart. They have to be so because the detailed contents of the learner models could not be known in advance.

Of course, every player of the pole and cart game can interpret differently terms like ‘quickly’, ‘fairly quickly’, ‘moderate’, ‘almost vertical’, etc., but a convention to evaluate the responses to the questionnaire had to be established beforehand. So the verbalization built by PACMOD (Section 6.1.3) was used to determine whether a rule matched a question.

A further pair of questions asked the participants to assess the quality of the suggested strategy and the accuracy of the learner models, and to describe any changes they considered should be done to improve it. After that, the participants were prompted to use PACMOD’s editing facilities (Section 6.1.3) to carry out their modifications to the strategy or learner model.

A time limit of thirty minutes was set in Experiment 1 for the participants to inspect and modify the strategy or learner model. This contrasts with the conditions of Study 3, where the participants could spend as much time as they wished inspecting the set of rules and answering the questionnaire. The system used for the experiment included all the enhancements suggested by the results from the previous study.

6.3.2 Results

Inspecting the suggested strategy

The record of the interaction between the participants in the **Expert** group and PACMOD indicates they spend from eighteen minutes to half an hour reviewing the set of rules and answering the questionnaire. A summary of the interaction, shown in Table 6.2, indicates that they did use the facilities to run and verbalize most of the rules that constitute the suggested strategy (the exception being participants 9, 16 and 11).

**Table 6.2:** Summary of how much the **Expert** group used the facilities to run and verbalize rules

Facility	Participant									
	9	11	12	13	14	16	21	24	26	28
Run	0	0	8	8	8	0	7	8	8	8
Verbalize	0	7	6	7	8	3	7	4	7	8

All the participants in the **Expert** group got right the first two questions in the questionnaire. Their responses to the next three questions are presented in Table 6.3, together with the correct answers. Their responses were correct in 7 out of 30 cases; the correct answer was accompanied by other (incorrect) responses in 11 more cases; and there were 12 completely wrong responses.

**Table 6.3:** Answers of the **Expert** group to questions 3, 4 and 5 of the questionnaire.

Question (correct answer)	Participant									
	9	11	12	13	14	16	21	24	26	28
3 (1)	1, 3	3, 1	1	3, 1	3	1, 3	3	1	1	1
4 (5)	See note <sup>a</sup>	5, 1	5	5, 1	3, 5	5, 7	7	1, 5	1, 3, 7	7
5 (6)	See note <sup>b</sup>	7, 2	7	7, 1	2, 4, 6	6	8	2, 6	2, 4, 7	6

<sup>a</sup> “To the left, max. speed pushing.”

<sup>b</sup> “Pushing very slowly to the left.”

**Inspecting the learner models**

The participants in the **Model** group spent from eight minutes to half an hour reviewing their learner models and answering the questionnaire. Their use of the facilities for justifying, running and verbalizing rules is summarised in Table 6.4. All the participants in the group, except Participant 27, answered correctly the first two questions of the questionnaire, and Table 6.5 shows the accuracy of their responses to the remaining questions (together with summary information about their learner models).

**Table 6.4:** Summary of how much the **Model** group used the facilities to justify, run and verbalized rules.

Facility	Participant									
	1	4	6	10	15	19	20	23	27	29
Justify	100	0	27	12	22	17	0	0	42	8
Run	100	0	45	0	100	100	100	0	100	0
Verbalize	60	0	9	25	11	0	0	0	92	0



**Table 6.5:** Summary of learner models and responses of the **Model** group. ‘✓’ indicates a correct answer; ‘✗’ represents an incorrect one; ‘✓-’ indicates an incomplete correct answer; ‘✓✗’ means the answer is mixed; ‘n’ indicates all answers are near misses (see discussion in Section 6.3.3); ‘✓n’ represents and correct answer plus near misses; and ‘-’ indicates no answer.

	Participant									
	1	4	6	10	15	19	20	23	27	29
Model prop.										
<i>push-left</i> rules	3	5	4	4	5	7	5	7	6	5
<i>push-right</i> rules	2	7	7	4	4	5	3	4	6	7
Preconditions										
per rule	2.2	4.4	3.2	3.6	3.6	3.5	3.6	3.4	4.2	4.3
Question										
3	✓	✓-	✓	✓-	✓	✓	✗n	✗	✓✗	✓
4	n	✗	n	✓✗	✓	n	n	✗	n	n
5	n	-	✓n	✗	n	✗	✓-	✓✗	✓	✗

**Editing the suggested strategy**

Eight of the participants in the **Expert** group considered the suggested strategy was *good*; another one regarded it as *very good*; and one more did not respond the question—the other two options were *bad* and *very bad*. Their comments and suggested improvements were along the following lines:

1. Make the preconditions of the first two rules more specific, either to make their role clearer or to limit their application for more ‘subtle control’ of the pole and cart.
2. Include a high level of description of the strategy, in the form of a ‘game plan’.
3. The cases where the cart is near to leaving the window require special treatment, not given by the strategy as it stands.
4. Eradicate ‘ambiguities’ in the verbalizations, particularly in relation to the description of velocities. The meaning of *between* is unclear for extreme velocities with different directions.
5. The last two rules seem to be ‘the wrong way round’ (opposite action).

6. There are cases where no rule would fire.
7. Specify the magnitude of the force, not only its direction.

Unfortunately, most of the participants ran out of time to make any changes to the suggested strategy. In fact, only three participants actually modified the strategy, spending between three and seven minutes in this task. Consistent with his comments, Participant 16 included three new rules at the top of the strategy: to cope with the ideal situation in which the pole and cart are both static, the pole is vertical and the cart is centred (*wait* action); and to give special treatment to situations in which the cart is very close to a border and moving quickly towards it (action of pushing the opposite direction to movement). Also consistent with his comments, Participant 24 restricted the preconditions on pole velocity (in all the rules, not only the first two ones), and refined the precondition on cart position in Rule 5. Participant 26 made changes to the preconditions on pole angle in Rules 3 to 6, and added preconditions on cart position to the first two rules, apparently to consider “give up” behaviour. He did not exchange the actions of Rules 7 and 8 though, despite his comments that they actions were swapped.

### Editing the learner models

Following a similar trend to the **Expert** group, seven of the participants in the **Model** group regarded their learner models as *mostly accurate*, and another one did not answer the question. However, two of the participants (19 and 29) considered their models were *mostly inaccurate*—the other two options available were *very accurate* and *quite inaccurate*. The former suggested a longer playing period to collect data for the construction of the learner model, as well as ‘greater weighting on the last few minutes’. Other participants commented they could not visualise their own strategy in terms of rules. Other comments on the models were along the following lines:

1. Models are too big; many of the rules appear to be very similar, and many rules are too vague.
2. Models should be symmetrical—in the sense that for each rule with a *push-left* action there should be a corresponding rule with a *push-right* action and symmetrical preconditions, and vice versa (see Section 4.4.2 for more details).
3. Include “give up” in the strategy model.
4. Make smaller the ranges for cart velocities.

5. Include rule for precondition *X*.
6. Consider the fact that a new control run could be terminated, and another one started, prior to a crash<sup>9</sup>.

More members of the **Model** group did some editing of their learner models than members of the **Expert** group, yet about half the former ran out of time to do any editing at all. The record of the interaction of Participants 1 and 27 with PACMOD shows that they did some changes to their models, even if they finally cancelled their changes, on purpose or by mistake. The changes that the remaining participants made to their models can be summarised as follows:

**Participant 6** narrowed ranges of pole angle and velocity, and eliminated preconditions on cart position (Figures D.3 and D.4). His intention appears to be decreasing the similarity between rules by making them more specific, and making the learner model more symmetrical (see the annotated transcription of the editing activity in Section D.1.2).

**Participant 15** added a couple of rules at the top of the model, to maintain control when the pole is almost vertical and the cart is almost centred but they are moving slowly in opposite directions (left and clockwise, or right and anticlockwise, respectively). He also changed the action of one rule to push in the opposite direction (Figures D.6 and D.7).

**Participant 19** modified ranges and deleted rules, to make the learner model more symmetrical (Figures D.8 and D.9).

**Participant 23** changed ranges of cart position and velocity of some rules at the bottom of the model (Figures D.11 and D.12). Although he did not give any comments on his model, he apparently tried to make the rules sort of complementary.

### 6.3.3 Discussion

The set of rules presented to the **Expert** group in the experiment (Figure 6.9) has a neater appearance than the set of rules used in Study 3 (Figure 6.6). In particular, it is symmetrical. It can be expected that the presentation of this set of rules would produce less confusion among learners. In contrast, the learner models presented to the members of the **Model** group were generally complex: asymmetrical, with more rules and more preconditions per rule (see Appendix D).

---

<sup>9</sup>This was because PACMOD allowed it, but it should not have done it.

The fact that only one in twenty participants answered the first two questions in the questionnaire incorrectly shows they understood the arrow notation used for presenting actions, with no difference between the two groups. The responses to the next three questions, which tested understanding of the way the preconditions of rules were displayed, were more varied. A strict evaluation of the **Expert** group's responses discards the 11 partially (in)correct answers—i.e. consisting of the correct answer plus one or more incorrect ones—and compares only 7 completely right answers against 12 totally wrong ones<sup>10</sup>. The result would suggest that the **Expert** group did not fully understand the presentation of the rule preconditions, but it will not discard a broad understanding of it.

A close inspection of Table 6.3 reveals that all incorrect answers but three (two from Participant 9 and one from Participant 21) are *near misses*, in the sense that three out of four of the preconditions of the rules given as answers satisfy the descriptions given in the questions. For example, Rule 3 is almost a correct response to Question 1: it fails only to satisfy the requirement of the pole 'falling fairly quickly to the right', because the verbalization of the rule characterises the pole as moving with a velocity 'between very slow anticlockwise and very slow clockwise'. It can be said that whether Rule 3 is a proper answer to Question 1 or not is a matter of individual judgement, since the interpretation of 'fairly quickly' can vary from one person to another. Furthermore, the interpretation of terms like 'fairly quickly' by an individual may depend on the set of rules presented to him; for example, a speed may be perceived as 'slow' if another faster speed appears in the strategy. From a point of view that takes into account all these problems, the results for the **Expert** group are much more encouraging and positive.

The **Model** group gave as many correct answers as the **Expert** group (7), but less partially (in)correct answers (7), and more wrong ones (15). These results suggest the **Model** group had more difficulty in comprehending the learner models than the **Expert** group in comprehending the predefined set of rules—as expected, since the learner models were more complex. Nevertheless, the high number of responses containing near misses (10), the existence of 3 incomplete but right answers, and the handicaps on the evaluation posed by the problems described in the above paragraph, suggest that the participants in the **Model** group acquired a broad understanding of the presentation of their learner models.

Table 6.4 shows the **Model** group used less the interactive facilities of PACMOD than the **Expert** group, in spite of the fact that the learner models presented to the former were in general

<sup>10</sup>In fact, an even stricter evaluation could regard the 11 partially (in)correct answers as plain wrong, but I believe such a level of strictness make sense only for experiments better controlled than this one.

more complex than the set of rules presented to the latter. A possible explanation for this is that the participants in the **Model** group got overwhelmed by the complexity of what they saw on the screen, and they had little motivation to explore it. Their little use of the interactive facilities may be related to the higher number of errors in their answers to the questionnaire, but there is no further evidence to support this claim.

Some participants commented positively on the textual explanations of rules, in spite of the fact that these are no more than verbose translations of the symbolic encoding of the rules. The graphical presentation of a rule is less accurate than its symbolic-numeric counterpart, but is still too concrete, lacking the generality of expressions in natural language—e.g. compare the expression ‘push to the left if the pole is tilted to the left, or slightly tilted to the right, and the cart is moving to the right not so slowly’ with the graphical and symbolic description of Rule 2 in Figure 6.2. According to some theories, this is not simply a problem of the graphical language chosen to express the learner models, but an inherent property of any graphical language (Stenning & Oberlander, 1995).

Finally, although many of the participants in the experiment ran out of time to make any changes to the set of rules presented to them, the remaining participants made changes to the rules that are generally consistent with their comments, to the point that the latter can be thought of as explanations of the former.

In conclusion, the results of Experiment 1 that pertain to the effectiveness of the interface to the learner models support the claim that learners can get an overall understanding, with some level of detail, of the conventions used in PACMOD to present the learner models.

## 6.4 Obstacles to model comprehensibility

A number of obstacles remain in the way for learners to comprehend their learner models. Some of these problems are related to the presentation of the models by PACMOD; others, have their roots in the nature of the models, their structure and content.

Presenting ranges of angles by arcs, and ranges of positions by boxes is a straightforward choice that happens to be effective too. In contrast, the adequacy of presenting ranges of velocities by means of animations is less clear. Part of the problem seems to reside in that a couple of animations represents a range of velocities only in an indirect way: each animation stands for a single velocity, and it is the couple of velocities which define a range of velocities. An arc or box, by contrast, represent a range of angles or positions in a more direct way, reducing the cognitive load on the learner—somehow he can *see* the range in the arc or box

but has to work out the final referent of the pair of animations. This is not an isolated result: recent research has shown that processing animations and dealing with interactive facilities can impose a heavy load on users of a computer system (Jones, 2000; Lowe, 2000; Narayanan & Hegarty, 2000).

An obstacle to understanding that has deeper roots is the lack of an explicit rationale in the learner models. All the learner can see is a set of disconnected rules which *he* has to integrate in a coherent interpretation: a structure of intentions, situations and plans. In the same way as graphical languages may be inherently concrete, unstructured set of rules may be inherently difficult to comprehend (unless they are rather small, and the learner models produced along this research generally are not). This is a caveat of systems of productions rules known as their *opacity* (Barr & Feigenbaum, 1981).

In summary, there are a number of reasons why learner models constructed with the method described in Chapter 4 should be easier to comprehend when presented using the graphical interface described in this chapter: they are presented in the same modality as the domain task; they are more modular, because the precondition of rules are now disjoint; and they can be explored, or even modified, interactively. The results of a pair of studies show that learners can get the overall idea and some degree of detailed understanding of their learner models, even if greater and more detailed comprehension is rare. Two plausible reasons of this difficulty are the lack of explicit rationale in the learner models and, to a lesser extent, the cognitive load imposed by the animations.

## Chapter 7

# Testing the effects of participative learner modelling

“Pooh,” he said, “where did you find that pole?”

Pooh looked at the pole in his hands.

“I just found it,” he said. “I thought it ought to be useful. I just picked it up.”

“Poo,” said Christopher Robin solemnly, “the Expedition is over. You have found the North Pole!”

“Oh!” said Pooh.

A. A. Milne (1926). *Winnie-The-Pooh*.

This chapter presents the results of two experiments designed to test the hypotheses laid down in Section 2.2 on the effects on learner behaviour of inspecting their learner models:

- increased ability to articulate accurate domain knowledge;
- increased ability to use domain knowledge in flexible ways, particularly on novel tasks and conditions; and
- a initial decrease in performance of the original skill.

First, it instantiates the original hypotheses to the domain of controlling the pole and cart; then it describes the design and implementation of the experiments; and finally it presents and discusses the results of the experiments.

## 7.1 The effects contextualised

The main effect on learners of inspecting their learner models, hypothesised in Section 2.2, is the construction and reinforcement (reactivation) of declarative domain knowledge and meta-knowledge: knowledge that can be articulated and otherwise employed in flexible ways, particularly to cope with novel situations. Such knowledge contrasts with the more task-specific, efficient but rigid procedural knowledge. A sensorimotor task like controlling the pole and cart, on the other hand, becomes mostly automatic after a reasonable amount of practice, imposing little demands on short-term memory and attention (Sections 2.1.1 and 3.4). It is because of this that the skill of controlling the pole and cart has been called ‘subcognitive’ (Michie et al., 1990). In consequence, an increase in declarative knowledge cannot be expected to have a positive impact on performance: on the contrary, a negative effect can take place, due to interference of the new and activated declarative knowledge with precompiled and more efficient procedural knowledge.

The domain of the pole and cart admits a variety of tasks other than controlling the device. For example, explaining the strategy used to control the device; judging the difficulty in control of starting with the device in a particular state; elaborating a plan to cope with a given situation; controlling the device using a different user interface (e.g. with a different visualisation of the state of the pole and cart or a different input device); coaching a novice player and recommending a course of action; etc. These type of peripheral tasks benefit more from a flexible representation of domain knowledge than the main task of controlling the device, and hence improvements can be anticipated in novice performance at these tasks after inspecting the learner model. It can be argued, however, that novice controllers of the pole and cart still rely on a fair amount of declarative knowledge to control the device, which they can apply to the peripheral tasks as well. The question in this case is whether inspecting their learner models makes this knowledge more readily accessible and better organised, with consequent improvements in performance. In other words, a positive effect of inspecting the learner model in performance at the peripheral tasks should be apparent even in this case.

The experiments described below focus on testing for changes in the performance of novices at controlling the pole and cart and four related tasks subsequent to them inspecting and editing their learner models; changes that can be ascribed to the novices acquiring new or reinforcing their existing declarative knowledge. The predictions are as follows.

1. They should get better at reporting their own controlling strategies; that is to say, self-reports of their strategy should match their behaviour better.



2. They should be capable of using their knowledge of controlling the pole and cart to judge better the relative importance for controlling purposes of the variables that determine the state of the device.
3. They should evaluate more accurately the difficulty in control of starting with the device in a given state.
4. They may perform worse than before at the task of controlling the pole and cart using the same interface, but they should exhibit better transfer of their skill to the task of controlling the device using a different user interface.

Reporting knowledge demands conscious access to it. Knowledge accessible in this way is regarded as the archetypical example of explicit (conceptual) knowledge (Dienes & Perner, 1999). Controlling the pole and cart with either interface demands the same kind of access to knowledge; transfer of the skill of controlling with the pictorial interface to controlling with the control-panel interface depends on identical elements of knowledge required by both tasks (Anderson & Singley, 1993). Judging the relative importance of the state variables and the difficulty of starting control with the device in a given state requires knowledge to be explicit, accessible and flexible, yet its owner may not be able to express her knowledge in any external notation (Karmiloff-Smith, 1996).

## **7.2 Design and implementation of the experiments**

Two experiments were designed to test the effect of inspecting the learner model in the domain of controlling the pole and cart. These have been referred to as Experiments 1a and 1b in previous chapters. Both experiments consist of three stages: baseline setting, condition and testing (Figure 7.1; Appendix C).

### **7.2.1 Baseline setting**

In this stage the participants were first asked to fill in a short questionnaire with background information (age, gender, preferred sports, etc.). Afterwards, they were asked to control the pole and cart for a total time of six minutes, split into two periods: a familiarisation period of one minute and a playing period of five minutes. The initial states of the pole and cart were selected from the semi-random and hard-displaced sets, respectively (Section 3.3).

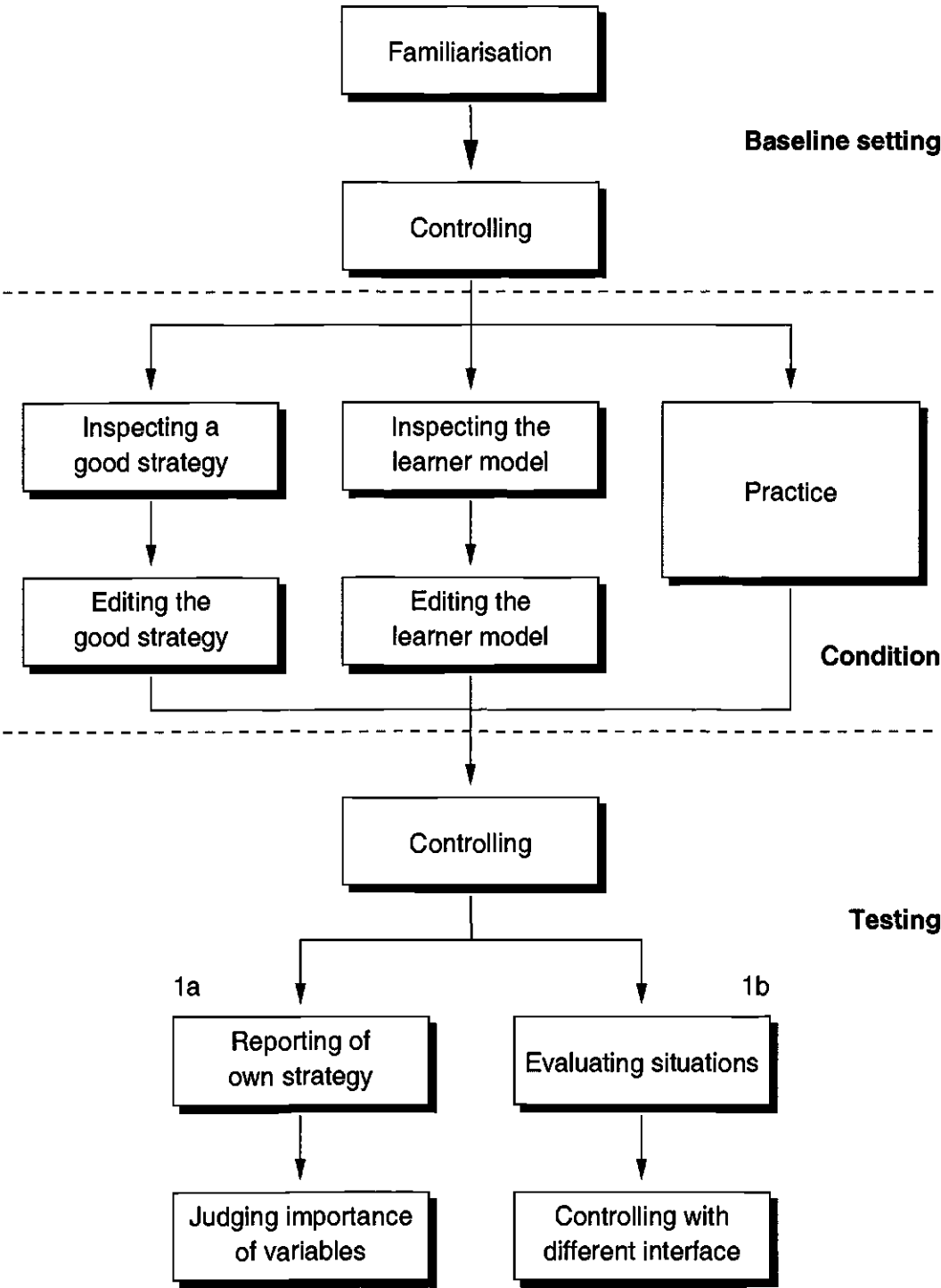


Figure 7.1: Overall structure of Experiments 1a and 1b.

### 7.2.2 Condition

In this stage the participants were split into three groups: **Expert**, **Model** and **Practice**. Each group was requested to perform a different task:

**Expert** A ‘good strategy’ was suggested to the participants in this group for them first to inspect and then to modify (Table 5.1 and Figure 6.9).

**Model** The learner model, constructed and tuned for presentation on the lines described in Chapters 4 and 6, was presented to the participants in this group for them first to inspect and then to modify (Appendix D).

**Practice** The participants in this group were asked to practice their control of the pole and cart with the aim of getting thoroughly familiar with the behaviour of the device. The initial states of the device for practicing were selected from the semi-random set.

The graphical interface described in Chapter 6 was used for presenting both the predefined strategy and the learner models; a brief user manual of the interface, tailored to each condition, was given as reference material (Appendix C). The strategy or model was displayed in view-only mode first, and the participants were asked to fill in a questionnaire to test their understanding of it. Afterwards, the participants were asked to express their disagreements with the quality of the strategy or the accuracy of the learner model. Finally, they were prompted to modify the strategy or model accordingly.

A total of thirty minutes was allocated for the inspection and modification tasks together, in both predefined strategy and learner model conditions, whereas twenty minutes were allocated for practice<sup>1</sup>.

### 7.2.3 Testing

All the participants were asked to control the pole and cart again, for five minutes. Afterwards, each group was split in half, and each subgroup undertook a different set of testing tasks, corresponding to Experiments 1a and 1b.

**1a** The participants in this experiment were requested to describe their own strategy for controlling the pole and cart (Section 7.7) and to rank the variables that define the state of the device according to their relative importance for controlling purposes (Section 7.8).

---

<sup>1</sup>There were two reasons to give less time for practice than for inspecting the learner models and predefined strategy. The first reason was that getting familiar with the graphical interface to the set of rules required some time, unnecessary for practice. The second reason was to avoid participants getting bored if they had too much practice in a relatively simple task.

1b The participants in this experiment were first presented with short simulations of the pole and cart device and asked to evaluate the difficulty of controlling the device starting from the state at the end of the simulation (Section 7.9). Afterwards, they were requested to play the pole and cart game for another five minutes, this time using a graphical interface that mimicked a control panel (Figure 7.2 and Section 7.6).

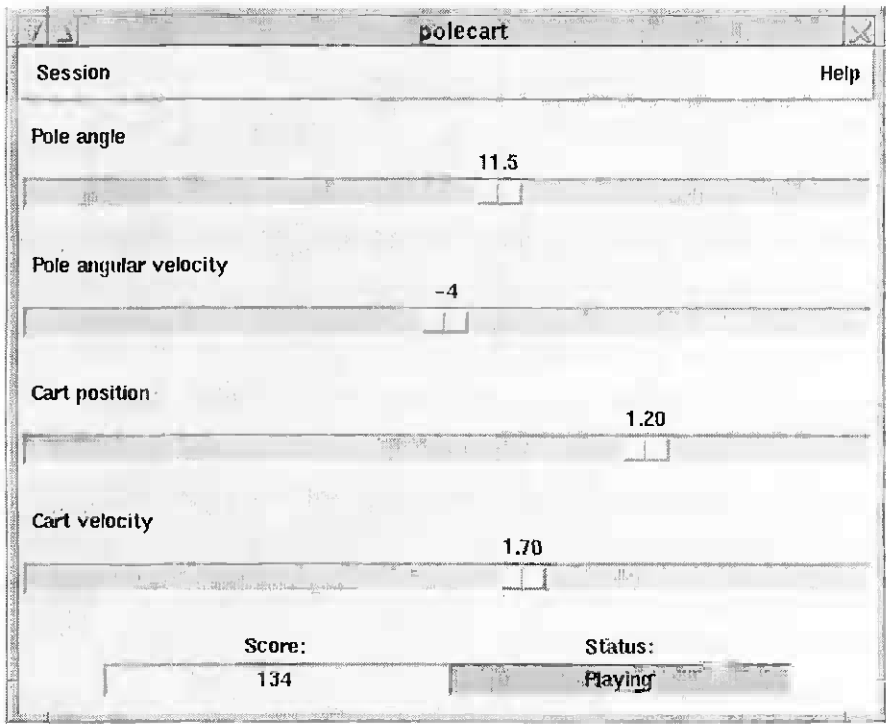


Figure 7.2: Control-panel interface to the simulator of the pole and cart.

The first task in this stage for Experiment 1a was aimed at assessing the ability of the participants in articulating their knowledge of their controlling strategies. There are a number of design possibilities for such a task. The simplest approach is to simply ask the participants to describe their strategy in their own terms, which has also the advantage of giving the participants freedom to express themselves in the best way they can conceive. However, it has at least two disadvantages: it does not give the participants any support for elaborating their answers, and makes the answers difficult to evaluate. Another approach is to design a restricted language for expressing strategies; one that facilitates the evaluation of the descriptions. A disadvantage of this approach is that participants have to be trained in the use of such a language, and hence they have to take time to learn it. A third approach is to lead the generation of descriptions through questions that encourage the provision of detailed descriptions. The disadvantage of

this approach is that some descriptions may become too forced and, consequently, rather inaccurate.

The three approaches described above were combined into a single design—one that kept their best properties and removed their disadvantages—along the following lines:

1. Ask the participants to describe their strategies in free form first and then by specifying the situations in which each one of the three possible actions are taken.
2. Translate the responses to the more specific questions into the formal language used to describe the learner models, and check consistency with the free form description.
3. Evaluate the accuracy of the descriptions, once translated, by matching them to the behaviour of the participant in the controlling task.

#### 7.2.4 Implementation

The sample of participants for the experiments was a mixture of postgraduate students (most of them from within the Division of Informatics, but also a few from Ecology, Engineering, Chemistry and Physics) and undergraduate students, all volunteers. The average age of the participants was 25 years ( $min = 18$ ,  $max = 37$ ), although seven of them were 18 years old. Twenty three of the participants were male.

A total of fifteen people took part in each experiment; that means ten people for each one of the conditions in the **Condition** stage, but only five people for each combination of condition and test. The assignment of experiment and condition was done at random. The instructions for the experiment and the questionnaires were provided in written form to the participants, though they were allowed to ask any clarification questions they considered necessary.

### 7.3 Expected outcome

Given the short duration of the **Baseline setting** stage of the experiments, compared with the longer **Condition** stage, more practice in the latter should be the best way of improving the skill to control the pole and cart using the original pictorial interface (Figure 3.2).

Inspecting the learner model should provide the best support for novices to report accurately their control strategies. Inspecting an alternative strategy may be helpful too—through promoting reflection—but it can also bias the way they conceive their own strategy: they may borrow parts of the alternative strategy to explain their own tactics of control. Finally, prac-

tice should have a detrimental effect on novices' ability to explain their behaviour because it reinforces implicit procedural knowledge.

It is not clear whether inspecting a learner model should provide more leverage than inspecting any good strategy, or vice versa, for reasoning about the pole and cart and controlling the device using the control-panel interface. Practice of the original task, on the other hand, should lead to worse performance of these tasks—again, because it reinforces less flexible procedural knowledge. Moreover, practice strengthens associations between sensory cues and actions (Section 2.1.1), making a skill less transferable when the sensory cues change markedly. An effect in this direction has been observed by Michie et al. (1990), who report no transfer of the skill of controlling the pole and cart using a pictorial interface to controlling using a control panel interface.

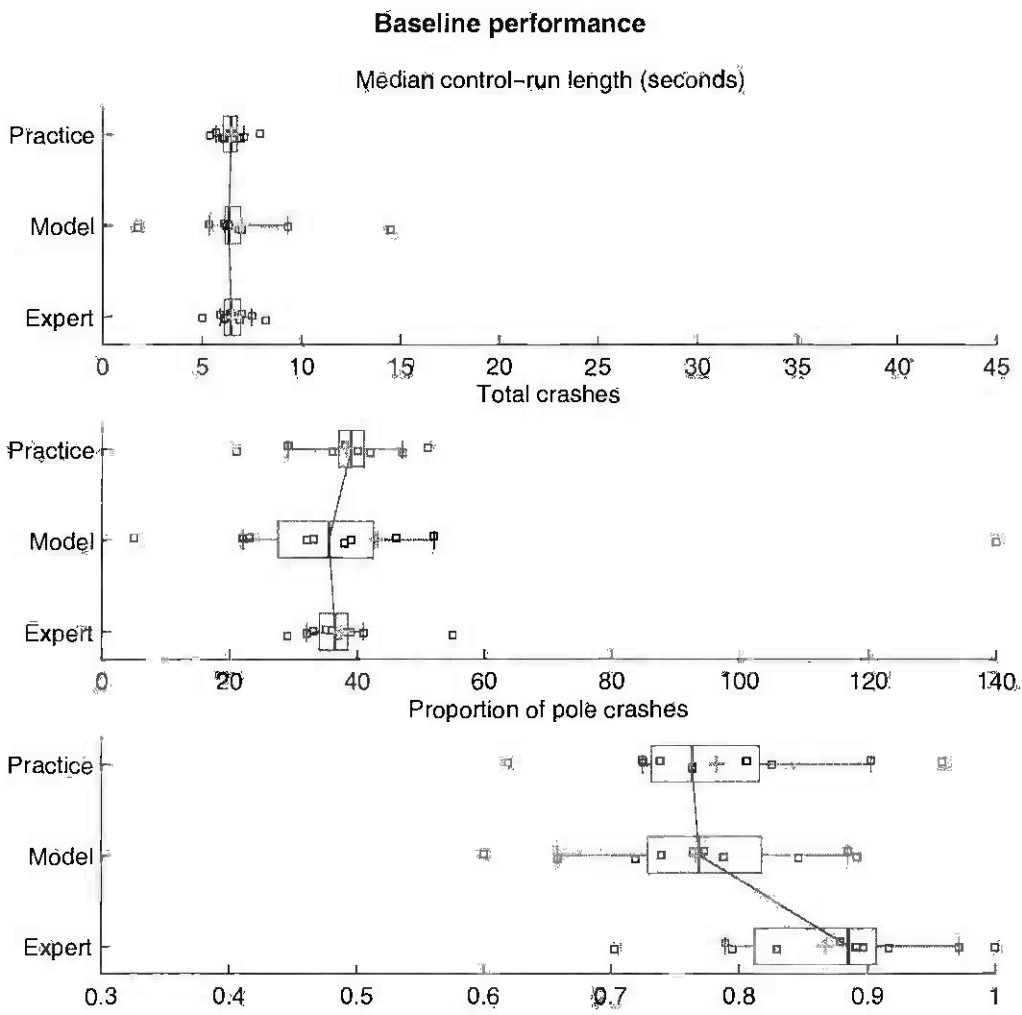
## 7.4 Baseline performance

The performance of the participants at the task of controlling the pole and cart is measure in terms of the same three variables used to characterise the performance of the artificial controllers in Section 5.2: median control-run length, total number of crashes and proportion of pole crashes—see Chapter 3 for a description of the task. Control-run length and total number of crashes are inversely correlated (i.e. the greater the number of crashes the shorter control-run length); however, the median is less sensitive to extreme values (probable outliers) and so adds some extra information. Novice controllers of the pole and cart tend to crash the pole more often than the cart, with the proportion of pole crashes tending to decrease with practice (Michie et al., 1990; Section 3.4). This is why the proportion of pole crashes is included in the characterisation of controller performance.

The baseline performance of the participants in both experiments is summarised in Figure 7.3. Two important facts are apparent from the graph:

1. The overall performance per condition was very similar in respect to control-run length and total number of crashes, but the proportion of pole crashes was higher among participants in the **Expert** group.
2. There were two clear outliers among the participants in **Model** group: one performed very well and the other very badly, in relation to the average.

The analysis of the data using a non-parametric statistical test of differences known as Kruskal-Wallis test (Table 7.1) shows no significant differences between the three groups in median



**Figure 7.3:** Baseline performance of the participants in Experiments 1a and 1b.

control-run length and total number of crashes. However, the same test suggests an important difference in the proportion of pole crashes ( $p < 0.057$ ). In conclusion, although it can be assumed there is no significant overall difference in performance between groups, the presence of the outliers and the difference in the proportion of pole crashes are warnings for the analysis henceforth.

### 7.5 Posterior performance

The participants in both experiments were asked to control the pole and cart for another five minutes immediately after finishing the tasks in the **Condition** stage—i.e. either inspecting and

**Table 7.1:** Summary of results of Kruskal-Wallis ANOVA of the baseline performance of the participants in Experiments 1a and 1b.

Kruskal-Wallis One-way ANOVA by condition				
	Mean ranks	Corrected for ties		
		$\chi^2$	D.F.	Sig.
control-run length	16.00, 15.45, 15.05	.0588	2	.9710
Number of crashes	14.80, 14.40, 17.30	.6397	2	.7263
Prop. of pole crashes	20.90, 12.30, 13.30	5.7097	2	.0576

editing their learner model or the suggested “good” strategy, or practicing their control of the pole and cart. The performance per group is summarised graphically in Figure 7.4, whereas the percentage of improvement per group is presented in Figure 7.5.

Figure 7.5 shows that improvement in performance within groups was highly variable, yet that each group’s overall performance improved in some degree can be appreciated by comparing Figure 7.4 with Figure 7.3. In particular, the group who got more practice at controlling the pole and cart exhibits the highest variability in improvement. The results of comparing posterior performances between groups using the Kruskal-Wallis test (Table 7.2) indicate that differences between groups are more significant than before (cf. Table 7.1). In comparison, a parametric one-way ANOVA is less conclusive in respect to the difference on the proportion of pole crashes but suggests an important difference in control-run length (a one-tailed<sup>2</sup> significance level of  $p < 0.07$ ). The difference between the results of both analysis is due to the presence of the three larger values of control-run length for the **Practice** group, which Kruskal-Wallis disregard (as outliers) but a parametric ANOVA takes greatly into account.

In conclusion, there is some indication of greater improvement in performance due to practice but the results are far from conclusive. This is so because improvement has high variability within each group.

### 7.6 Performance on the transfer task

The last task in the **Testing** stage of Experiment 1b was to control the pole and cart using a user interface that resembles a control panel (Figure 7.2). Instead of showing a drawing of the

---

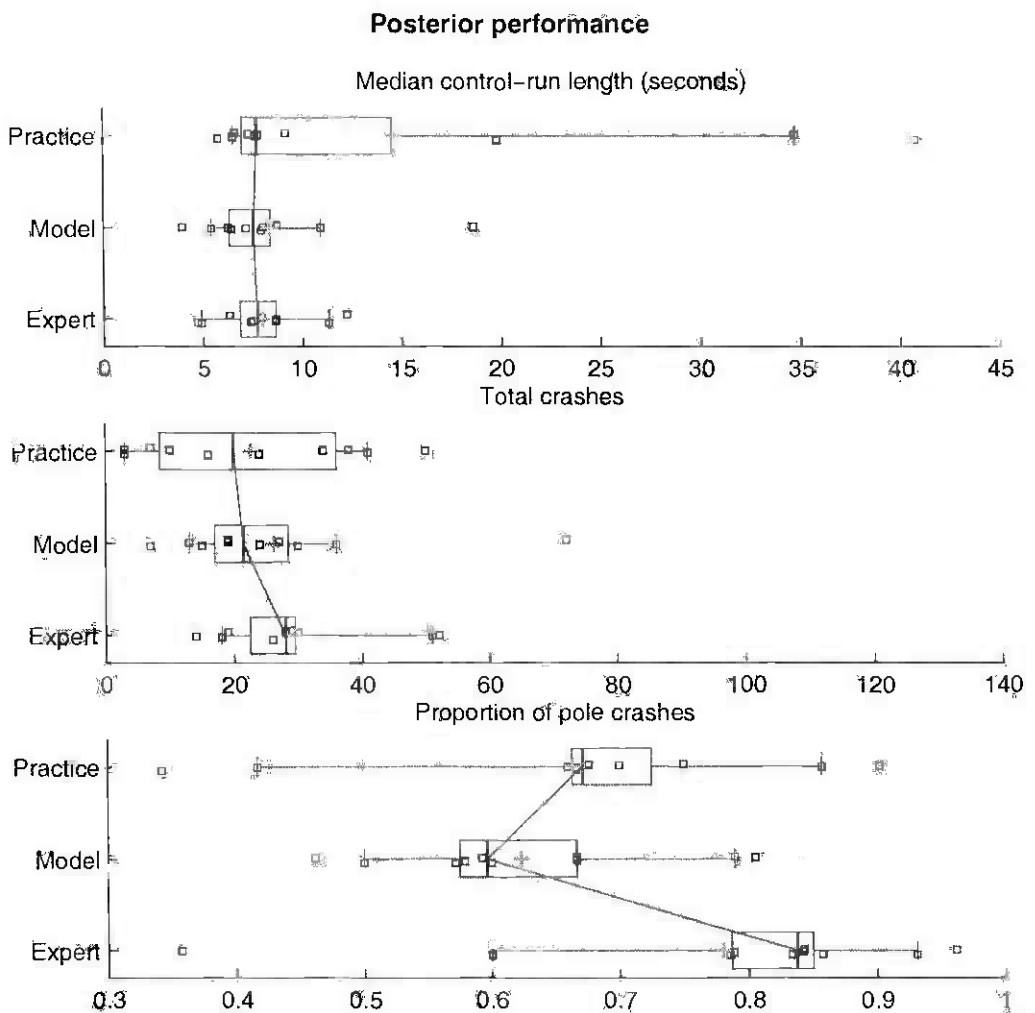
<sup>2</sup>Comparing the performance of the Practice group to the others’ performance using one-tailed tests is justified because the improvement in its performance was expected to be the greatest (Sections 7.3 and 7.3). Differences in performance between the other two groups, if any, have to be analysed using more strict two-tailed tests.



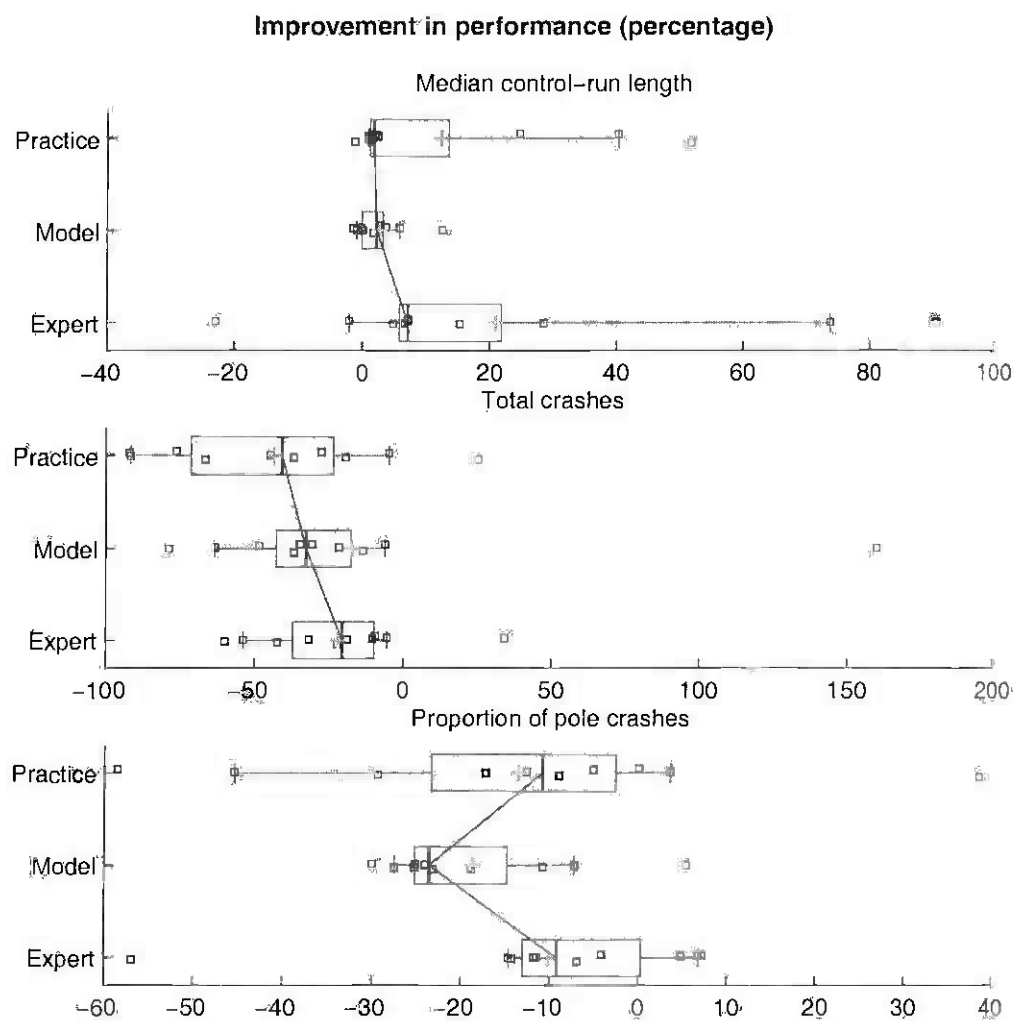
pole and cart device, as in the pictorial interface (Figure 3.2), the values of position, angle and velocities are displayed in their numerical form and as positions of sliders in scales.

Graphical summaries of the performance of the participants using this interface are shown in Figure 7.6. A comparison of this performance with their baseline performance (a subset of the performance shown in Figure 7.3) shows that the performance of the **Expert** group improved in the transfer task, suggesting that they were able to transfer their experience on the original task to the new task. The performance of the **Practice** and **Model** group, on the other hand, did not get noticeably better.

A increased performance on the transfer task was expected from both the **Expert** and **Model** groups, compared to the third group. A one-tailed Kruskal-Wallis test supports the



**Figure 7.4:** Posterior performance of the participants in Experiments 1a and 1b.

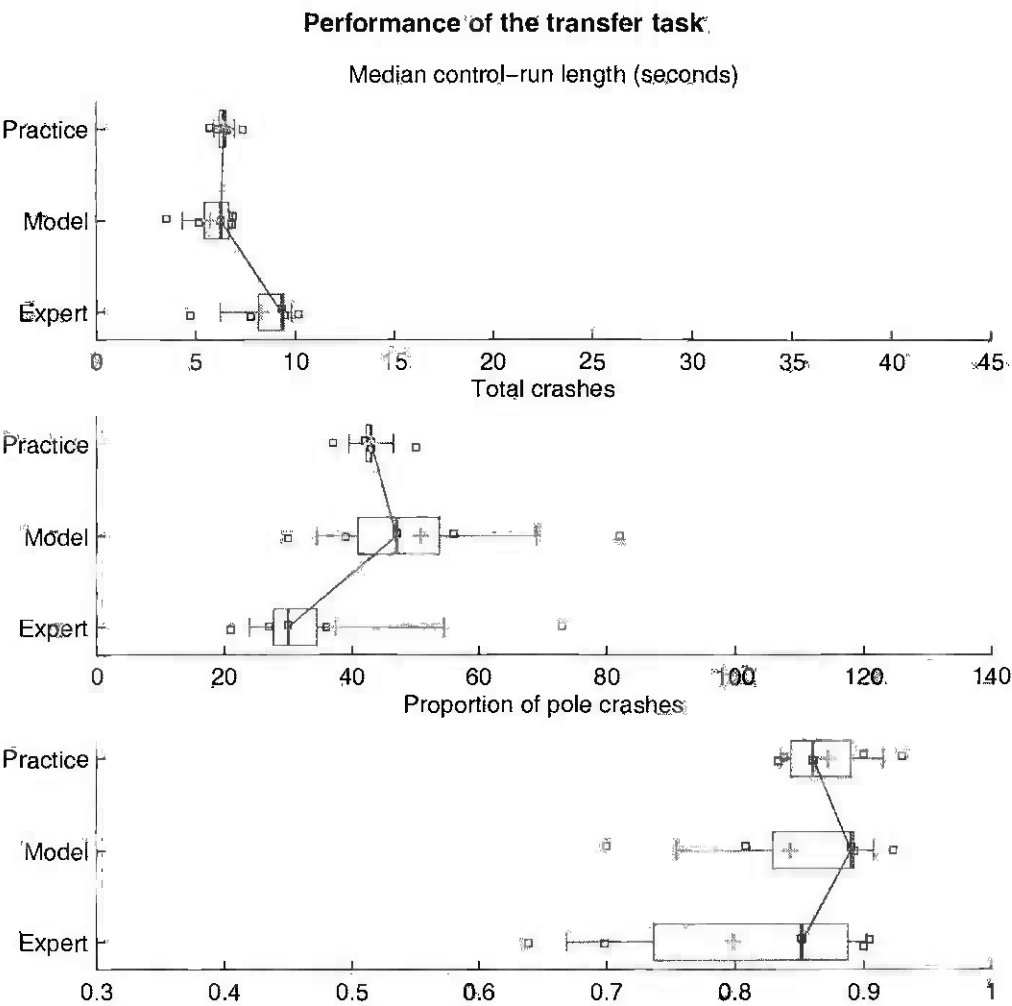


**Figure 7.5:** Percentage of improvement between baseline and posterior performance among the participants in Experiments 1a and 1b. The percentage is calculated as  $100 \times (\text{posterior performance} - \text{baseline performance}) / \text{baseline performance}$ .

**Table 7.2:** Summary of results of Kruskal-Wallis ANOVA of the posterior performance of the participants in Experiments 1a and 1b.

Kruskal-Wallis One-way ANOVA by condition				
	Mean ranks	Corrected for ties $\chi^2$	D.F.	Sig.
control-run length	14.85, 13.95, 17.70	.9893	2	.6098
Number of crashes	18.05, 14.85, 13.60	1.3621	2	.5061
Prop. of pole crashes	20.80, 10.95, 14.75	6.3854	2	.0411

original hypothesis for the **Expert** group, in relation to control-run length ( $\chi^2 = 4.02$ ,  $df = 2$ ,  $p < 0.10$ ) and total number of crashes ( $\chi^2 = 3.255$ ,  $df = 2$ ,  $p < 0.10$ ). The results are then good and disappointing at the same time.



**Figure 7.6:** Performance of the transfer task by the participants in Experiment 1b.

## 7.7 Articulation of the playing strategy

### 7.7.1 The questions

The first task in the **Testing** stage of Experiment 1a was to answer the following four questions:

1. Try to give an overall description of any strategy you follow for keeping the cart inside the window and avoiding the pole falling over the cart.

2. Under which conditions of the pole and cart do you execute an action of pushing to the right?
3. Under which conditions of the pole and cart do you prefer to wait for a change of conditions?
4. Under which conditions of the pole and cart do you execute an action of pushing to the left?

The participants were allowed to choose their own way of expressing the answers and most of them used natural language; some included drawings too. The difference between the first question and the others is that the latter ask specifically for the constraints the state of the pole and cart must satisfy for the participants to execute a certain action, whereas the former asks the participants to provide an overall description of their strategy.

### 7.7.2 Translation of the answers

The answers to the questions above were translated to the language of the learner models following a sort of inverted procedure to the one used to generate textual explanations of rules in the learner models (Section 6.1.3). The conventions followed when translating the answers, and the problems then faced, are described below.

#### Conventions

The language used by the participants to answer even the most specific questions was richer than the one available to the system; the language employed to answer the first question was also rather different. Besides the descriptors used by the system to generate the textual explanations of rules, such as ‘very fast clockwise’ and ‘close to the centre’, the participants used other, more or less equivalent descriptors, such as ‘close to stationary’ and ‘nearly/roughly/almost vertical’. The equivalence relation between the new and old descriptors had to be established before proceeding to their translation into the formal language of the learner models.

Other descriptors used by the participants have no equivalent among the ones used by the system. For example, the participants frequently described the state of the pole and cart as either ‘falling’, ‘toppling’, ‘raising’, ‘balanced’ or ‘going to go off the screen’. For each one of these higher level descriptors equivalent expressions, in the form of combinations of more basic descriptors, had to be defined. On other occasions, the participants used expressions like ‘beginning to fall’, ‘beginning to move’, ‘slowing down’ and ‘when starting a control run’,

which somewhat involve time. These expressions describe a situation in the context of an ongoing sequence of events—e.g. if the pole is “beginning to fall,” that means it was not falling a short while ago—and need to be “decontextualised” before ascribing them a corresponding combination of basic descriptors. Finally, a few participants included seemingly redundant information in their responses, introduced with expressions like ‘specially’ and ‘if it is possible’. For example,

When the pole is nearly vertical, and the cart is nearly stationary, *specially* when the slight angle and the slight cart velocity are in the same direction. (Participant 1; my emphasis)

The interpretation of such an extra information was that it was meant to put emphasis on a set of conditions (a rhetoric device) and so it was translated as additional rules, in spite of the fact that these superfluous rules have no effect on the predicting power of the translated description.

### Problems

As can be inferred from the conventions explained above, a great deal of interpretation was involved when translating the responses from the participants into a formal language, and decisions had to be taken when tradeoffs were identified. In summary, the translation of the textual descriptions provided by the participants into the formal language of the learner models was driven by the principles of accuracy and parsimony (as few assumptions as possible) but taking into account the consistency and utility of the result—some examples below clarify the latter. Among the problems encountered, the following deserve special mention: provision of numeric values, seemingly too strong conditions, long translations of short statements, slips and different levels of description.

**Specific values** Occasionally a participant included a concrete numerical value for a parameter (e.g. ‘when the pole is more than 10 degrees to the right’) accompanied by qualitative descriptions (e.g. ‘nearly vertical’). In these cases, a decision had to be taken about how to define the quantitative meaning of the qualitative description: either in terms of the numeric value mentioned in the answer or in terms of the numeric values used by the system when generating the textual explanations of rules—the latter used in most other cases. The tradeoff here is between consistency across translations of all answers and accuracy in translating a specific answer. The decision taken for these cases favoured accuracy over consistency.

**Strong conditions** Also occasionally a participant described a situation as the pole ‘standing still’ or the cart being ‘centred’ or ‘stationary’. The direct translations of these expressions into formal language are so restrictive (i.e. the velocity and angle of the pole are *exactly* zero, and the velocity or position of the cart are *equal to* zero, respectively) that only a very small set of possible situations meet their requirements, no one of them to be encountered in practice. A looser translation would increase the chances for the description to be useful in practice, but at the expense of lessening the accuracy of the translation. The decision taken for these cases favoured practical utility over accuracy.

**Long translations of short statements** The translation of some general descriptions, like ‘moving quickly’ without specifying the direction of movement, include more rules and preconditions than translations of more detailed descriptions, like ‘moving quickly to the left’—in this particular case two rules and two preconditions versus one rule and one precondition. This fact is paradoxical, to say the least, since it is arguable that more specific descriptions provide more information than their more generic counterparts. That it happens points out to the fact that in the translation process important information is lost and new information is added.

A careful and impartial translation in which unavoidable assumptions are made explicit should reduce distortion and the incorporation of noise (extraneous information) into the result (Ericsson & Simon, 1984), or at least should make distortion and noise uniformly distributed. At the same time, careful analysis of the context should allow the recovery of information that otherwise will be lost by a straightforward translation. For the case in hand, whether the long translation of a short statement like ‘moving quickly’ is right or wrong depends on the *intention* of the participant explaining her strategy: whether she actually *means* moving quickly in *either* direction, or she just fails to realise that the general phrase does not completely describe the target situation. The decision taken for these cases, nevertheless, was to assume as little as possible besides what was explicit in each participant’s response, yet maintaining a certain level of congruence between the pieces that composed the translated version of the whole answer.

**Slips** Small inconsistencies in the answers to the more specific questions appeared from time to time. For example, minor differences between the descriptions of situations for pushing in either direction, such as in

**Push-right:** If the pole is roughly vertical, the cart is moving slowly and is heading away from the middle of the screen.

**Push-left:** If the pole is almost vertical and the cart is moving slowly away from the middle of the screen *to the right*. (Participant 15; my emphasis)

and in

**Push-right:** When travelling left and the pole is *beginning* to move clockwise. When travelling right and the pole is falling clockwise.

**Push-left:** When moving to the right and the pole is moving anti-clockwise. When moving to the left and the pole is falling anti-clockwise. (Participant 14; my emphasis)

In the first case, the preconditions for pushing to the left are more restrictive than the preconditions for pushing to the right; in fact, the latter contains the former as a special case! A slip by the participant, who failed to specify ‘to the left’ at the end of the conditions for pushing right, seems a reasonable explanation of the situation. The situation is reversed in the answers of Participant 14: the conditions for pushing to the right are more restrictive than for pushing to the left. In this case, however, there is no contradiction between both sets of conditions; furthermore, the conditions for pushing to the left are more complete. Consequently, there are no good reasons to assume a slip in the second part of the answer, yet it is not clear whether we should take ‘beginning to move’ as a sloppy expression for ‘moving’ in the first part. The best option in this case might be translating the conditions as they are, even if that results in an asymmetrical strategy. In general, the context of the answers to the whole set of questions should help to disambiguate whether the difference can be attributed to a slip or to a more serious misconception.

**Different levels of description** Most of the participants expressed their answers to the open question about their controlling strategy using a language very different to the one used by the system in its textual explanations of rules. Among the more important differences are descriptions of

- goals and plans to achieve them (e.g. ‘I generally aimed to bring the cart to a stable state around the middle of the screen’),
- situations and actions contextualised in a sequence of events (e.g. ‘Accelerate quickly[,] wait until the pole is at a certain angle then brake heavily, accelerate a little to straighten the pole, then brake very slowly’),

- frequency of keystrokes (e.g. ‘I used short taps on the keys to move the cart left and right’),
- “brokenness” of the pole (e.g. ‘As the line gets “broken” as degree changes ... The number of break points increases’),
- judgements (e.g. ‘The best thing is to do small short movements ... Long fast [movements] ... are difficult to control’),
- high level “cognitive” strategies (e.g. ‘Try to foresee and predict the result of the actions’) and
- overall conceptualisation of the task (e.g. ‘This was done by concentrating on keeping balancing the [cart] *under* the pole, as opposed to keeping the pole over the cart’).

On the one hand, the different level of description used for answering the first question and the latter ones makes comparing them harder. On the other hand, having two rather different descriptions of the same strategy increased the confidence in the faithfulness of the translation, specially when a good level of consistency was achieved.

### 7.7.3 Evaluation

The accuracy of the formal descriptions of controlling strategies, derived from the textual descriptions provided by the participants in the study, was measured in terms of how well they matched the way the participants controlled the pole and cart at the beginning of the **Testing** stage, just before writing their descriptions—see Section 7.5 for more information on their performance. The procedure to compare each formal description to the corresponding behaviour was the following:

1. The data representing behaviour was prepared so as to produce a learner model and used as training data to give each rule a weight (see Section 4.4).
2. The same data was used as testing data to measure the predicting power of the description of the strategy. Two measures of predicting power were taken: power to predict any of the actions in the data, including *wait* actions, and power to predict *push-left* and *push-right* actions only.

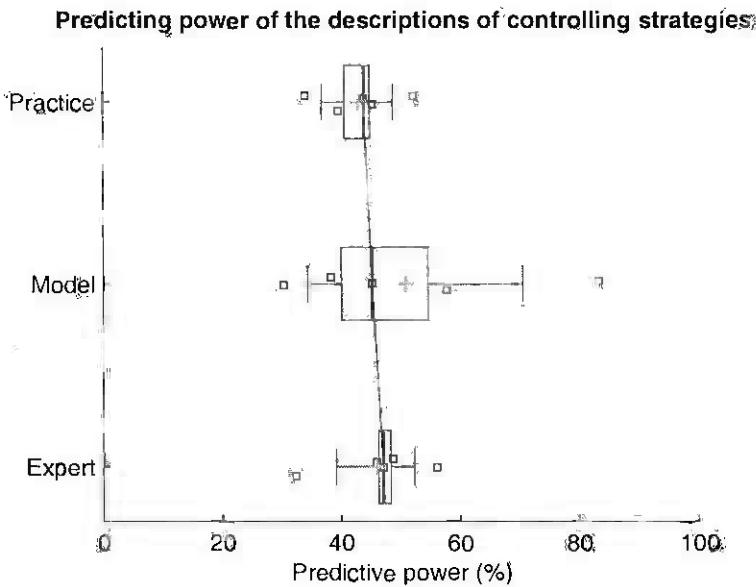


- 3. The gain in predicting power by the descriptions, compared to betting in favour of the action most frequently executed, was calculated using the formula

$$\frac{P(\text{description}) - P(\text{most frequent action})}{P(\text{most frequent action})}$$

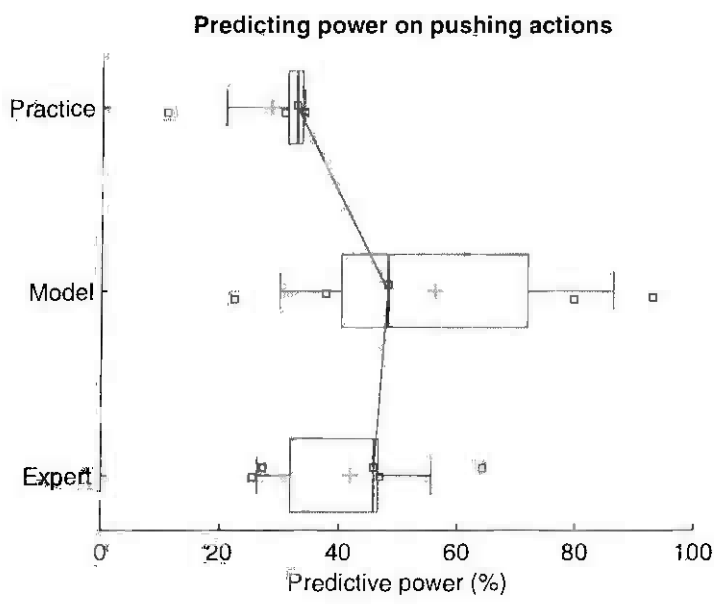
where *P* stands for predicting power.

The overall and restricted predicting power of the descriptions are presented graphically in Figures 7.7 and 7.8, respectively. The first graph shows that descriptions from participants in the **Practice** group are slightly less powerful at predicting behaviour than descriptions from the other two groups. The difference between groups is increased when predicting power is measured only on *push-left* and *push-right* actions, as shown in the second figure, with the direction of this difference as expected—i.e. the **Practice** group is the worst (Section 7.3). A one-tailed Kruskal-Wallis ANOVA gives these latter results a significance of  $p < 0.104$  ( $df = 2$ ,  $\chi^2 = 3.14$ ); one-tailed parametric ANOVA gives them a higher significance ( $p < 0.07$ ,  $df = 2$ ,  $F = 2.3712$ ) because the difference between means is higher.

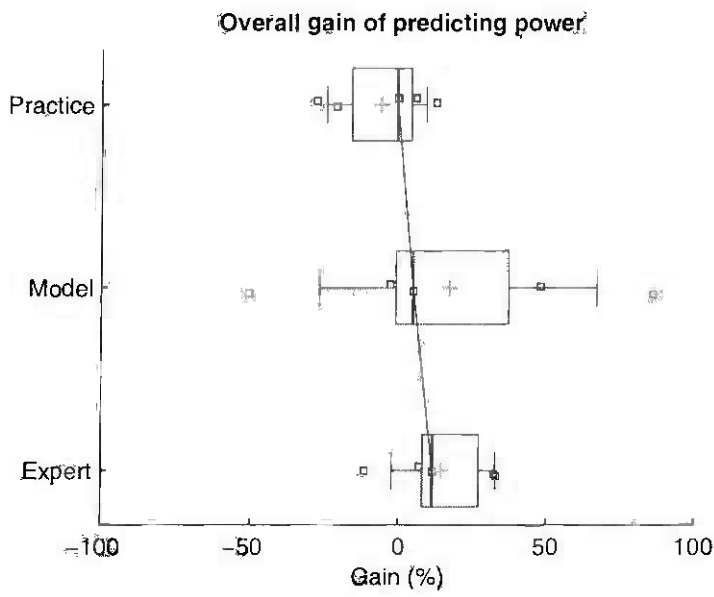


**Figure 7.7:** Overall predicting power of the descriptions provided by the participants in Experiment 1a.

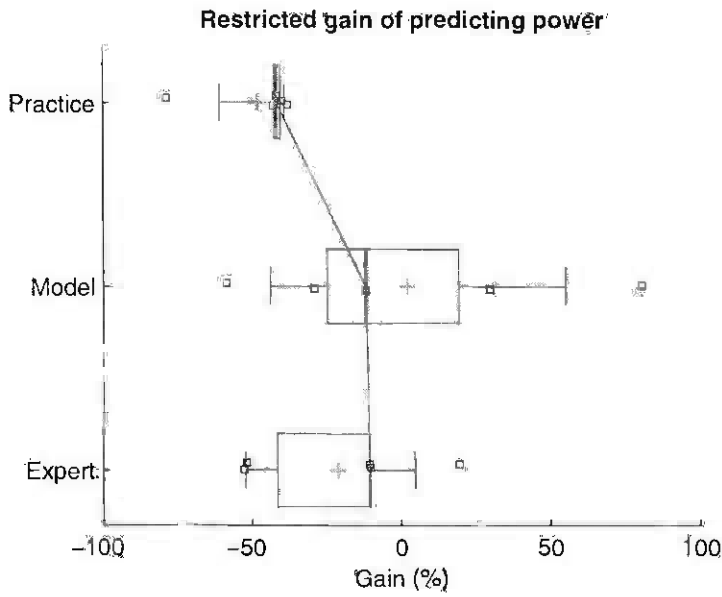
The gain in predicting power over betting in favour of the action most frequently executed is shown graphically in Figures 7.9 and 7.10. These results show the same tendencies as the ones in the previous figures. In addition, they show that the descriptions given by the participants provide a substantial gain over guessing only in a few cases, specially when restricted to actions other than *wait*.



**Figure 7.8:** Predicting power of the descriptions provided by the participants in Experiment 1a restricted to *push-left* and *push-right* actions only.



**Figure 7.9:** Gain in overall predicting power by the descriptions provided by the participants in Experiment 1a when compared to guessing in favour of the most frequent action.



**Figure 7.10:** Gain in predicting power by the descriptions provided by the participants in Experiment 1a, restricted to *push-left* and *push-right* actions, when compared to guessing in favour of the most frequent action.

## 7.8 Ranking of state variables

The last task in the **Testing** stage of Experiment 1a was to rank the variables that define the state of the pole and cart. The question was posed to the participants in the following manner:

Rank the following properties of the pole and cart according to their relative importance for controlling purposes. Give them numbers from 1 (most important) to 4 (less important). Assign the same number to all properties you regard as equally important.

Cart position [ ]      Cart velocity [ ]      Pole angle [ ]      Pole velocity [ ]

The answers given by the participants are presented in Table 7.3.

A variable is more or less important for controlling purposes to the extent that it determines the behaviour of a controller. Consequently, one should be able to estimate the relative importance of a variable for a controller by analysing the correspondence between the different values of the variable and the actions executed by the controller. This is precisely what is done when a model of the controller's strategy is induced from the controller's behaviour.

Following this reasoning, the relative importance of the state variables for each participant can be estimated by counting the number of rules in the model of the participant with pre-

**Table 7.3:** Ranks of the state variables, as reported by the participants in Experiment 1a.

Condition	Participant	Cart		Pole	
		Position	Velocity	Angle	Velocity
Expert	9	4	2	1	3
	11	2	3	4	1
	12	3	4	1	2
	13	4	2	3	1
	14	3	1	2	3
Model	1	4	2	1	3
	4	1	2	1	3
	6	4	2	1	3
	10	4	3	1	2
	15	3	2	1	3
Practice	2	4	1	2	3
	3	3	1	1	2
	5	3	2	1	4
	7	2	1	1	3
	8	1	1	2	2

conditions involving each one of the state variables; each rule weighed by its score, calculated using Equation (4.2). The basic assumption behind this estimation is that the model is a faithful interpretation of the participant’s strategy<sup>3</sup>—an issue discussed in detail in Chapter 5.

Since the strategy of the participants could have changed from the **Baseline setting** to the **Testing** stage, their behaviour in the latter has to be taken into account. Consequently, a model of each participant’s strategy was induced from their behaviour when controlling the pole and cart in the **Testing** stage (Section 7.5), and the weighed count of rules in this model with preconditions involving a variable was used as an estimator of the variable’s importance. The result of this process, after transforming the numbers obtained into ranks, is presented in Table 7.4. The relationship between the importance of the variables calculated from the learner models and the ranking provided by the participants was estimated using the Spearman’s  $\rho$  coefficient of correlation (Conover, 1999, Ch. 5). Although the results are mixed, they somewhat favour the **Model** group, specially if compared to the **Expert** group (Figure 7.11).

<sup>3</sup>It is also important that all the models were generated following the same procedure, and the rules in each model have disjoint preconditions.

**Table 7.4:** Ranking of the state variables estimated from the number of occurrences of each variable in the learner models of each participant in Experiment 1a.

Condition	Participant	Cart		Pole	
		Position	Velocity	Angle	Velocity
Expert	9	3	4	1.5	1.5
	11	2	4	1	3
	12	3	4	1.5	1.5
	13	3	4	1	2
	14	3	4	1	2
Model	1	3.5	3.5	1.5	1.5
	4	4	3	1.5	1.5
	6	3	4	1	2
	10	3	3	1	3
	15	3.5	3.5	1	2
Practice	2	3	4	1	2
	3	4	3	1	2
	5	4	3	1	2
	7	4	3	1	2
	8	3	2	1	4

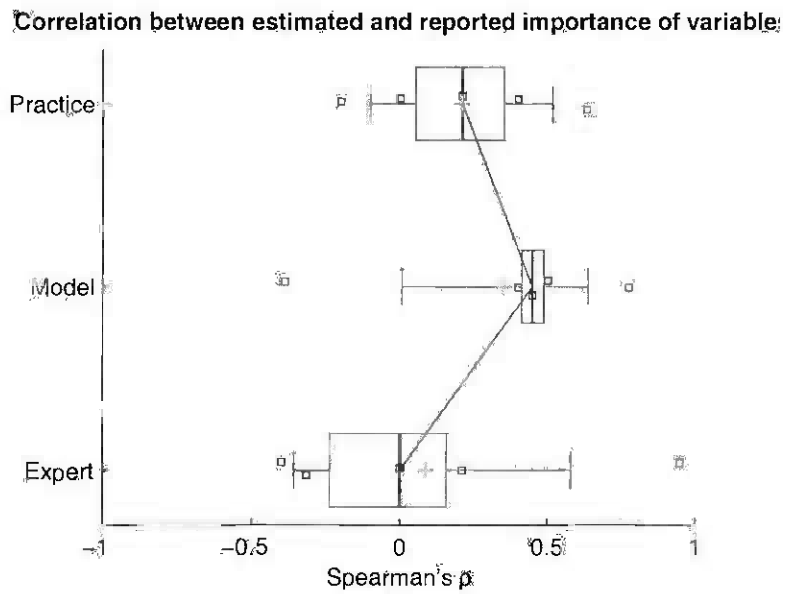
7.9 Ranking of states of the pole and cart

The first task in the **Testing** stage of Experiment 1b was evaluating the difficulty in controlling the pole and cart device starting from a state chosen from a predefined set. The goal was to measure the ability of the participants to judge the controlling difficulty by observation and reasoning, as opposed to simply reacting to a situation.

The space of states of the pole and cart can be thought of as divided into a set of “boxes” (Michie & Chambers, 1968). Within each box, the pole is either falling or raising, and the cart is either leaving the window or coming in to the centre. The states of the pole and cart were chosen from these boxes to make them representative of generic configurations of the device; the full list is shown in Table 7.5.

The task was posed to the participants in the experiment in the following way:

A sequence of ten short animations of the pole and cart will be displayed on the screen. Each animation will end with the pole and cart in different conditions.



**Figure 7.11:** Correlation between the estimated and reported importance of the states variables per participant per condition, measured as the Spearman's  $\rho$  between the two rankings.

Please grade the final conditions of each animation using a scale from 1 to 7, with 1 standing for the pole and cart being in conditions totally under control and 7 standing for conditions completely out of control.

Under control    1   2   3   4   5   6   7   Out of control

4.154

Each animation lasted for 400 ms only (but could be repeated at will); its initial state was chosen such that the animation ended in one of the states shown in Table 7.5. The degrees of difficulty ascribed to each animation's final state by the participants in the experiment are summarised graphically in Figure 7.12. It can be seen that the participants strongly agreed when given extreme values to the easiest and hardest cases (states 3 and 8, respectively). The responses were more varied for the remaining states, notably states 6, 7 and 9.

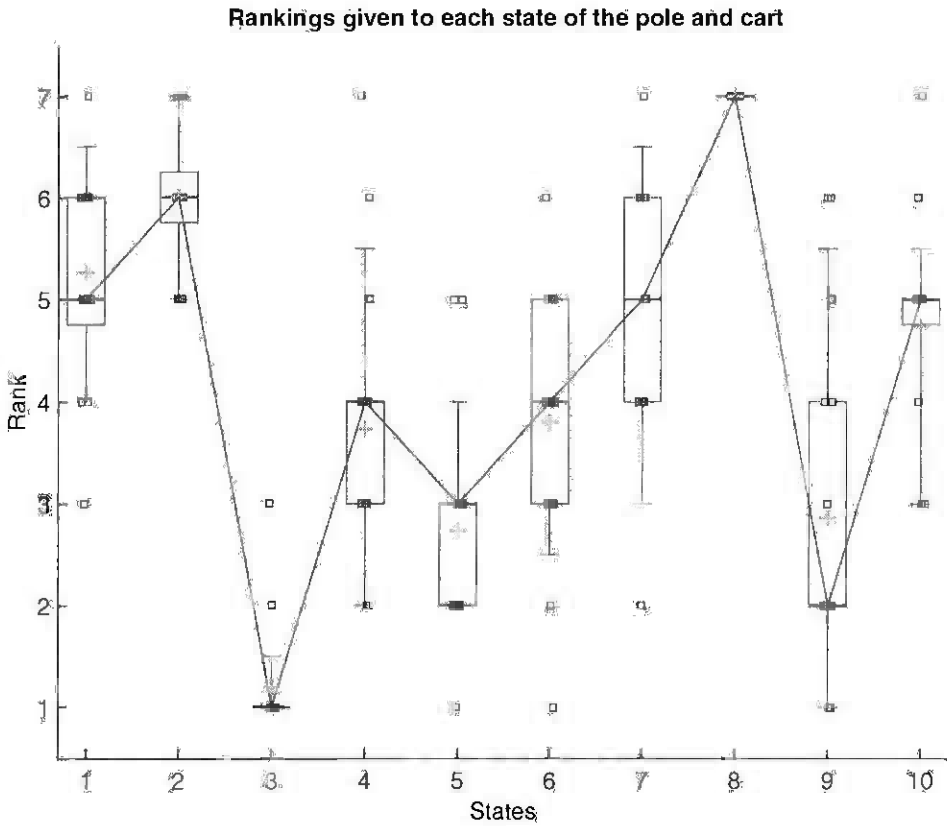
In order to estimate how difficult it would be for each participant to control the pole and cart starting from each one of the states listed in Table 7.5, the performance of each participant was simulated by executing the learner model inferred from their behaviour at controlling the pole and cart in the **Testing** stage, and the control-run length was used as an estimator of difficulty. Control-run lengths were then transformed into ranks and compared to the ranks

**Table 7.5:** States of the pole and cart that were ranked by the participants in Experiment 1b.

#	Cart		Pole		Description
	Position (m)	Velocity (m/s)	Angle (rad)	Velocity (rad/s)	
1	1.25	-1.86	0.43	1.49	Cart approaches the centre while pole falls to the other side.
2	-1.48	-1.83	-0.43	-1.49	Cart leaves and pole falls towards the same side.
3	-0.14	0.13	0.01	-0.05	Cart almost centred and pole rather vertical, both moving slowly.
4	-1.48	-1.83	-0.29	0.85	Cart leaves by one side while pole raises from it.
5	-1.25	1.83	0.29	-0.85	Cart approaches the centre while pole raises from it.
6	1.48	1.86	-0.29	0.85	Cart leaves by one side while pole raises from the centre.
7	1.48	1.86	-0.43	-1.49	Cart leaves by one side while pole falls to the other.
8	2.12	3.04	0.88	2.93	Cart almost out while pole topples to the same side.
9	1.25	-1.86	0.29	-0.85	Cart approaches the centre and pole raises to the same side.
10	1.25	-1.83	-0.43	-1.49	Cart approaches the centre while pole falls towards it.

provided by the participants<sup>4</sup>, using Spearman's  $\rho$  as a measure of correlation. The results, presented graphically in Figure 7.13, show a high degree of correlation between the estimated levels of difficulty and the participants' reckoning for the **Expert** and **Practice** groups, and a significantly lower correlation for the **Model** group ( $p < 0.013$ ,  $\chi^2 = 8.72$ ,  $df = 2$ ).

<sup>4</sup>The same rank transformation was applied both to the control-run length and the answers from the participants.



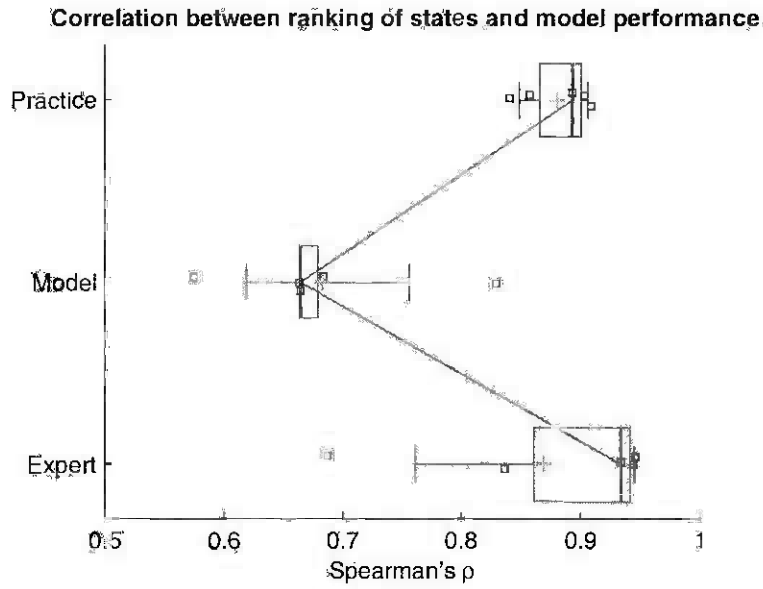
**Figure 7.12:** Rankings given by the participants in Experiment 1b to each one of the ten states of the pole and cart they had to judge.

### 7.10 Discussion

The **Practice** group’s posterior performance at the original task was somewhat better than the performance of the other groups, but without showing a clear advantage of practice over the other conditions. A plausible explanation of this result is that boredom and fatigue affected the **Practice** group at this point, after performing the same sort of task for nearly half an hour. In any case, this particular result is of minor relevance for the purposes of this research, more concerned about the effects of the other conditions.

Compared to the **Practice** group, both the **Model** and **Expert** group were better at reporting their strategy for controlling the pole and cart, specially if the evaluation discards *wait* actions—considered a source of noise in Chapters 4 and Chapter 5. The **Expert** group was good also at judging the difficulty of controlling the pole and cart starting from a given state and at the transfer task. The results of the **Model** group were the other way round: poor performance both at the transfer task and at ranking states by difficulty. Finally, the results for the





**Figure 7.13:** Correlation between the states' degree of difficulty, estimated and judged by the participants in Experiment 1b.

task of ranking variables by importance were very mixed, yet favouring the **Model** group.

A light-hearted summary of the results of the experiments is shown in Table 7.6—light-hearted in the sense that non-significant differences are nevertheless taken into account. Although the impression one can get from the table (and certainly from the full description of the experiments in previous sections) is that the results are mixed, overall they slightly favour the **Expert** group over the **Model** and **Practice** groups. The tasks where the **Expert** group performed particularly poorly were the original task of controlling the pole and cart with the pictorial interface and ranking the states variables according to their importance to control: the former was anyway expected to be dominated by the **Practice** group; the latter produced the most mixed results.

Looking back to Section 6.3, where an evaluation was presented of how much the participants understood their learner models and the predefined strategy, it appears that the greater difficulty in comprehending the learner models may have had some influence on the final outcome of the experiments. Better understanding of the hand-coded strategy by the **Expert** group could be a reason behind the difference between its poor performance at the original task in the **Testing** stage and its good performance at the transfer task and at ranking states by difficulty—the sort of effect that was expected for the **Model** group. If that were so, learner models that are easier to understand would produce similar effects on the **Model** group. However, it can also

**Table 7.6:** Summary of the results of Experiments 1a and 1b ('o.t.' stands for one-tailed significance).

Experiment	Task	Performance			<i>p</i>
		Best	Medium	Worst	
1a & 1b	Control (pictorial)	P	M	E	
1a	Reporting strategy	M	E	P	0.11 (o.t.)
1a	Ranking variables	M	P	E	no sig.
1b	Ranking states	E	P	M	0.013
1b	Control (c. panel)	E	P	M	0.10 (o.t.)

be the case that inspecting a "good" and clean strategy (as an instance of the general case of being instructed on how to perform the task based on carefully distilled knowledge) increases the flexibility of knowledge more than inspecting the learner model. In any event, the issue still remains of why the **Expert** group performed poorly at ranking variables.

It is suggestive that the members of the **Model** group performed somewhat better than the others precisely at the task of reporting their strategy: as if they were actually more *aware* of their strategy as a result of inspecting their learner models. The greater difficulty in understanding an unpolished and inconsistent learner model can be seen in this case as an incentive to *reflect* harder on one's own way of tackling a problem.

Table 7.6 shows also that the **Model** group performed the best in the tasks specific to Experiment 1a but the worst in the tasks specific to Experiment 1b; the performance of the **Expert** group follows a similar but reversed pattern: the best in Experiment 1b but poor in Experiment 1a. This variability contrasts with the more consistent performance of the **Practice** group across experiments. Even though it cannot be discarded such a variability can be due to the different nature of the tasks in each experiment, it can also be no more than a side product of the rather small number of participants in the experiments to match the number of different conditions (three) and tests (five). Differences in gender, age and background among the participants need also be considered as a plausible source of fluctuations in the results.

All this variability adds noise to the results and makes it harder to observe clear effects of the different conditions. The fact that despite the noise some trends are still appreciable makes the outcome of the experiments more interesting and suggestive of the potential effects on learners of inspecting and editing the learner models.

## Chapter 8

# Discussion and conclusions

At first, when people create or find a new model of teaching that works for some purpose, they're so thrilled they try to use it for everything. Our job is to provide some order—finding out what each model can do and building categories to help folks find the tools they need.

B. J. Joyce, E. Calhoun and D. Hopkins (1997). *Models of learning—tools for teaching*.

This dissertation started with a definition *participative learner modelling* as a process of learner modelling characterised by the learner playing an explicit and active role in it (Chapter 1). The term 'participative' is preferred because it has less implications for the goals of the participants and their interactions, as opposed to other terms, more semantically loaded, like 'collaborate', 'cooperate' and 'negotiate'.

The inspiration for participative learner modelling comes from several sources, notably the desire to use learner models as educational tools for prompting learners to reflect on and be more aware of their own knowledge. However, the effectiveness of the learner models to promote reflection and knowledge awareness has mostly been assumed, with little empirical evidence put forward to support it. One reason for this neglect can be that the issue is simply so obvious and the argument quite convincing:

Some way of ensuring that learners really do use the information offered as a basis for reflection is also desirable. This can be achieved by allowing the learner to edit their model . . . , but with reactions from the system to the student's actions in their student models, requiring them to justify their decisions if the system disagrees (Bull, 1997a, p. 80)

Another, perhaps more subtle reason can be that it is really hard to devise a way of assessing the amount of reflection and knowledge awareness induced in the learner by participating in the

modelling process. The research that has been described here is an exploration of participative learner modelling driven by the aim of deciding between the two reasons above in a laborious way: by implementing participative learner modelling in a suitable domain, designing an indirect test for reflection and awareness in that domain and putting it into practice.

Participative learner modelling has been applied to modelling learners of cognitive skills like speaking and writing in a second language, using a computerised text editor and doing numerical calculations. Not only are reflection and knowledge awareness important in this kind of domains; it can be expected also for it to occur naturally, to some extent, even if covert learner modelling is used instead of a participative one. However, that might make it more difficult to appreciate the effects on the learner of the latter. A domain centred around a perceptual-motor skill, on the other hand, can be a better choice for testing the effects of participative learner modelling, even if reflection and knowledge awareness are less relevant for the subject matter—or perhaps precisely because they are less relevant for it. In addition, implementing participative learner modelling for a new kind of domain serves as a test of the generality of the approach. Controlling the pole and cart is a good domain for this exploration because, besides being centred on a sensorimotor skill, it is simple enough to facilitate experimentation without being plain boring to learn (Chapter 3).

The method developed for constructing the learner models is an adaptation of a method for acquiring expert knowledge using machine learning (Chapter 4): it takes traces of the learner's attempts to control the pole and cart as input data, applies an algorithm for supervised rule induction and produces as output a set of production rules specifying the preconditions for the execution of each one of the actions available to the learner. The models are hence extracted directly from learner behaviour with a minimum of assumptions about the learners and the task; the same input data can be employed to justify the models to the eyes of the learners. Although it cannot be claimed that the structure of the learner models matches the cognitive structures used to represent a learner's knowledge of how to control the pole and cart, it has been shown that they can be used to predict learner behaviour with fair accuracy (Chapter 5).

Up to here the modelling method is learner-centred but covert; the learner is treated as a passive object of diagnosis, unaware of what is going on behind the scenes. The process is made participative when the learner models are open to learner inspection and modification (Chapter 6). The models are presented in the same (graphical) modality as the domain task, to make it easier for learners to interpret them in terms of their previous experience with the task. The content of the models is refined to increase its modularity, so that each individual component of a model can be understood more independently of the rest. Mechanisms are included to

deliver justifications, verbal explanations and executions of the contents of the models. Finally, but not less importantly, facilities are provided for learners to directly manipulate the contents of the models, from whole rules to their preconditions to their actions.

With all this apparatus in place, the last part of the research focused on testing for learner reflection and knowledge awareness that resulted from inspecting and editing the learner models. This was done by firstly interpreting reflection and awareness in terms of classical theories of human cognition to deduce changes in learner behaviour posterior to participative learner modelling (Chapter 2). The conclusion was that reflection and awareness should increase knowledge reportability, flexibility and transferability, with the overt consequences of learners becoming better at reporting their knowledge of the task and at using their knowledge in novel ways, in particular to carry out tasks in the domain other than the original task of controlling the pole and cart. Following from this conclusion, a couple of experiments were designed and implemented to test for the occurrence of the hypothesised changes in learner behaviour (Chapter 7).

The results of the experiments are interesting and suggestive, although mixed and far from conclusive. Overall, they support the idea that inspecting and editing a set of production rules which describe a strategy for controlling the pole and cart improves performance at related tasks in the same domain. However, better results were obtained when the set of rules *was not* the learner model but a predefined “good strategy”.

The following section expands on the discussion of the experimental results given in Section 7.10. A view of this research as an instance of a methodology for exploring and testing participative learner modelling is given in Section 8.2. The dissertation finally concludes in Section 8.3.

## 8.1 The effects of participative learner modelling

The results of the experiments described in Chapter 7 suggest inspecting the learner model helps learners to better articulate their knowledge of how they try to control the pole and cart. The results support the claim that inspecting the learner model prompts learners to reflect on their knowledge and increases their awareness of it. However, the difference is only appreciable when inspecting the learner model is compared with having more practice at the control task, but not when it is compared with inspecting a well designed strategy. Furthermore, the participants in the experiments that inspected the learner models performed poorly at other tasks that demanded flexible access to domain knowledge at the object-level (not at the meta-level),

specially if compared with the participants that inspected the predefined strategy. In fact, inspecting the learner model is related to slightly worse performance at these tasks than having more practice.

If the results were obtained under ideal experimental conditions, they would lead to the conclusion that inspecting the learner model—as an instance of participative learner modelling—actually promotes reflection that increases knowledge awareness, but it does not make domain knowledge more flexible, accessible (other than to meta-cognition) or transferable. However, there are a series of factors that qualify the results and prevent jumping to such a conclusion.

**Small sample size** First of all, it was noted in Chapter 7 that the number of participants in the experiments was rather small to cope with the number of conditions and tests. On the one hand, that stresses the significance of the results<sup>1</sup>: tendencies may become stronger as the sample size increases. On the other hand, it can be argued that either the benefits of participative learner modelling do not show up in the small scale (i.e. weak effects) or that Table 7.6 is a product of chance; whereas the former statement is hardly a defence of participative learner modelling, the latter cannot be cast aside (see Section 8.3).

**Short-term evaluation vs longer-term effects** Participative learner modelling in practice is part of a series of interactions between the learner and the system that span from a few hours to several weeks or months, or even years (Kay & Thomas, 1995). The beneficial effects of these interactions are hoped to last even longer. In contrast, my experiments lasted less than an hour and thirty minutes, and so it can be the case that participative learner modelling was simply not given enough time to produce its effects. Moreover, most people, if not everyone, start learning a new skill in “controlled mode” and it takes a certain amount of practice for the skill to become more or less automatic (Sections 2.1 and 2.1.1), even with a sensorimotor skill like controlling the pole and cart. Therefore, it is plausible than a longer experiment—e.g. several sections of practice-condition-practice before applying the tests—would benefit the participants that inspected the learner models and handicap the people that only got practice in a single task.

---

<sup>1</sup>The levels of significance reported in Table 7.6 were calculated using statistical methods that are non-parametric, considered particularly robust in cases of small samples and uncertainty about the shape of the population distribution (Conover, 1999). One-tailed tests are justified by the initial hypotheses (Sections 2.2, 7.1 and 7.3)

**Inadequate tests** Controlling the pole and cart using a control panel may be the sort of task that can be done better if one is taught a generic solution to the problem of control than if one is confronted with one's own way of tackling it. Besides, the skill is similar enough to control with a pictorial interface to share a fair amount of knowledge with it—specially if one has not practised the latter to the level of depending crucially on perceptual idiosyncrasies. It comes as no surprise then that controlling from a panel is done badly by a novice controller if he is deprived of inspecting a well designed strategy and not given enough practice to master the share elements of knowledge, but instead is presented with his own ineffective and inconsistent strategy. Ranking states of the device by difficulty to control can also benefit from being taught a sound strategy, as well as from episodic memories of past control sessions. Moreover, because the states to be ranked were presented as the last states of animations, guessing the difficulty in control could be accomplished by sensing the readiness to react to each state produced; readiness that gets fine tuned with practice.

Transferability and flexibility of knowledge can be tested in other ways, by means of tasks more different to controlling the pole and cart: from simple variations on the tasks discussed above such as describing verbally the states to be ranked, to disparate tasks like recognising replays on one's own performance, planning how to cope with a given situation and teaching a learning companion (Ramírez Uresti, 1998). Other desirable properties of knowledge, like the learner's confidence in it, should also be tested.

**Unfair conditions** The evaluation of the participants' understanding of the set of rules they were presented with, representing either their personalised learner model or a predefined strategy, showed the learner models were more difficult to comprehend. In fact, they were more complex than the predefined strategy: with more rules and more preconditions per rule; judged subjectively, they also appeared less coherent—the predefined strategy is presented in Figure 6.9; the learner models are included in Appendix D. With this disparity in mind, the results of the experiments can be interpreted as suggesting that the extra difficulty of inspecting more complex learner models did not stop the participants from reflecting on and becoming more aware of their knowledge but hindered any further benefits of the task.

Rather than being a fault in the design of the experiments or a property of the pole and cart domain, the additional difficulty in understanding the learner models in the experiments may be the manifestation of an intrinsic problem of participative learner modelling: learner models will hardly be as clear and distilled as an expert model or the model of an ideal student; on the contrary, they will tend to be incoherent, inconsistent, incomplete and tangled. Learners

will have a harder time with participative learner modelling than otherwise because it is a ‘hard cognitive task’—as (Barnard & Sandberg, 1996) categorised self-explanations.

**Inadequate learner models** Although most of the participants that inspected the learner models regarded them as ‘mostly accurate’ (Section 6.3.2), their responses can be interpreted as a byproduct of lack of understanding—after all, the next question was about their disagreements with the model, followed by a request to modify the models accordingly: if they had agreed, there was little else to do afterwards. Inaccurate models would have affected the evaluation of the performance at the tasks of ranking state variables and states of the pole and cart (Sections 7.8 and 7.9). The models were fairly accurate, though; at least for predicting learner actions (Section 5.3).

In Lee’s (1999) terms, the learner models produced in this research are closer to data laws than to “proper” models. They are predictive, but their explanatory powers are limited and they *represent* learners in a very narrow sense, since the modelling method cannot warrant any psychological plausibility of its outcome. Novice players of the pole and cart game seem to explicitly pursue goals (e.g. stopping the device from leaving the window, moving it slowly towards the centre), whereas their models are purely reactive: they *explain* learner behaviour as reactive performance, making explicit the preconditions for each one of the learner actions. Given the simple nature of the domain, these sort of models may be sufficient, in spite of not corresponding to the level of abstraction learners conceive the task of controlling the pole and cart (Section 7.7.2). After all, major theories of skill acquisition propose concomitant knowledge is encoded as a system of productions (Section 2.1.1), and some researchers have suggested that goals can lose their explicit character, becoming only implicit in the preconditions of these productions (Rasmussen, 1983).

The purpose of constructing learner models in this research was to present them to learners for inspection, critical appraisal, and subsequent modification. The models would be adequate as much as they could be presented in a comprehensible form to learners for them to identify themselves with. It is not clear the extent to which the lack of an explicit rationale in the models was an obstacle not only to their comprehension but also to learners’ sensation of being inspecting and tailoring *their* models—a condition that was stressed in the experimental material (Appendix C).

**Domain too simple** Controlling the pole and cart was chosen because of its simplicity and perceptual-motor character (Section 3.2). However, it can be argued that the domain is too



simple for reflection and knowledge awareness to have any positive effect on novices. It may be innocuous, or even harmful. Basically, controlling the pole and cart was not selected as a testing domain because it were thought participative learner modelling was useful in it, but the other way round: it was chosen because the domain was thought useful for exploring participative learner modelling (Section 3.2). A decrease in controlling performance after inspecting the learner model was likely but did not occur to a significant degree; its possible effects on the other tasks have been discussed in Sections 7.1 and 7.3, and in this section when the adequacy of the tests was considered.

**Wrong underlying theory** The theories summarised in Chapter 2 are some of several explanations of human cognition. Other theories can be assumed instead (e.g. Cheng, 1985; Dienes & Perner, 1999; Karmiloff-Smith, 1992; Sun et al., 1999) which could result in different hypotheses of changes in learner behaviour as a result of being involved in the learner modelling process (Section 2.3). Alternatively, a more detailed model of the acquisition of the skill to control the pole and cart on the basis of the classical theory—e.g. based on ACT-R/PM (Byrne & Anderson, 1998), EPIC (Meyer & Kieras, 1997a) or SOAR (Newell, 1990)—would probably lead to more accurate models and more specific predictions.

## 8.2 A global view of the trail

This dissertation describes a proof of concept for participative learner modelling in a new kind of domain: acquisition of sensorimotor skills. A number of decisions were taken in the course of the research motivated by the long standing aim of testing the effects on learner knowledge of participative learner modelling; effects that could be observed indirectly through changes in learner behaviour. Other researchers on participative learner modelling have worked with more cognitive domains and focused on other issues: the *scrutability* of the models and learner control (Cook & Kay, 1994; Kay, 1994b, 1997; Zapata-Rivera & Greer, 2000); learner collaboration for constructing the model and use of the model as a tool for communication and supporting learning (Ayala & Yano, 1996; Beck et al., 1997; Bull, 1997a,b; Bull & Broady, 1997; Bull & Ma, submitted; Bull & Shurville, 1999); and the learner-system interaction in participative learner modelling (Bull, 1997a; Bull & Pain, 1995; Dimitrova et al., 1999a,b).

Research on Artificial Intelligence in Education (AIED) has often been criticised for trying to solve problems created in the laboratory, with little connection with the real problems teachers confront each day. Natural reactions to this criticism—and signs of the maturity of

the field—have been recent attempts to put intelligent learning environments into practice by taking them ‘to school in the big city’ (Koedinger & Anderson, 1997) and developing useful systems that tackle complex domains (e.g. McCalla, 2000; Vassileva et al., 1999b). A similar approach in this research would have been choosing a domain like air traffic control (Lee & Anderson, 1997) or piloting a remotely operated vehicle (Roberts et al., 2000), both good examples of fairly complex skills with a strong sensorimotor component. However, a toy domain was selected, because it facilitated experimentation with an AIED technique. The approach is not new; on the contrary, it has a long-standing tradition in AIED (e.g. Burton & Brown, 1982), Artificial Intelligence (e.g. Winograd, 1972; see Russell & Norvig, 1995, p. 19 for other references) and Cognitive Psychology (e.g. Lebiere & West, 1999)<sup>2</sup>.

The same *minimalist* approach was followed throughout the research.

- The form of participative learner modelled investigated is inspectable (and modifiable) learner models, which is not only a unsophisticated way of involving the learner in the modelling task but also happens to be a stereotypical instance of participative learner modelling (maybe because of its simplicity).
- The learner models are simple sets of production rules, implementing a simplistic reactive conception of the domain task.
- The models are inferred directly from low-level traces of learner behaviour by means of supervised rule induction (a traditional machine learning technique), using a well-tested program (RIPPER, Cohen, 1995). The amount of background knowledge required to build the learner models is hence minimal.

The models constructed in this way are rather *static*, but that was all that was needed to make them available for inspection and modification in the studies comprised by the research. Although it is possible to construct a sequence of models of a single learner by considering overlapping windows of their behaviour, the modelling process will start from scratch every time, using the same low-level input data and the same basic amount of background knowledge; it will not update a previously constructed model, but rather discard it and construct a new one<sup>3</sup>.

---

<sup>2</sup>In fact, Lee & Anderson (1997) worked with a much simplified version of the air traffic control problem.

<sup>3</sup>It can be expected that two consecutive learner models in a sequence are rather similar, because of a certain amount of overlap in the training sets they are induced from; because they are produced using the same mechanisms; and because both training sets are generated by the same subject (appealing here to some sort of here to psychological continuity). Nevertheless, from a methodological point of view, those are the only guarantees available for the method to work incrementally, as it certainly has been shown it does (Morales & Pain, 1999).

- The interface to the learner models was designed to present the models using the same modality as the domain task was presented. Even though some explanatory facilities are provided, so the learner can ask for the justification of a rule, a verbal translation of it or its execution, special attention was paid to minimise them, so they could not be labelled covert instruction.
- The theory supporting the predictions tested in the experiments has been termed ‘the classical view of [human] cognitive architecture’ (Stillings et al., 1995, Ch. 2).

The last point may seem out of place. After all, the ‘classical view’ makes a good deal of assumptions about the human cognitive architecture which have been under attack by more parsimonious views of the human mind like the connectionist (Bechtel & Abrahamsen, 1991; Stillings et al., 1995, Sec. 2.10) and the non-representational (Brooks, 1991). However, it is much harder to infer the effects of participative learner modelling under the latter views, as they are still clambering their way into meta-cognition. The classical view becomes, then, the straight and safe road to follow.

A minimalist research method does not guarantee success, as the caveats discussed in the previous section clearly indicate; yet, it is easier to go back and elaborate the initial proposal than to make it simpler.

### 8.3 Conclusions

The investigation described here has been both an exploration of participative learner modelling, giving the learner an explicit and active role in the learner modelling process, and a quest for its effects on learner knowledge that are observable through changes in learner behaviour. The main contributions of the research are:

- i) the application of participative learner modelling to a new kind of domain, centred around the acquisition of sensorimotor skills;
- ii) a set of concrete hypotheses about the effects on learners’ knowledge of playing an active role in the modelling process;
- iii) initial empirical results of testing these hypotheses in a domain that involves acquiring a sensorimotor skill (controlling a pole on a cart); and
- iv) a proposal for a methodology for carrying out further investigations on these issues, exemplified by the work described in this dissertation.

The methodological issue has been discussed enough in the previous section. The remaining of this section focuses on the first three points.

### 8.3.1 A new kind of domain

This research is different from previous work on participative learner modelling in the time-constrained and highly dynamic nature of the domain, centred on the sensorimotor task of controlling a pole on a cart. This kind of domain has been in a minority in research in learner modelling in general, but it is gradually becoming a very important field of application for intelligent technologies for education in the near future, and hence for learner modelling.

The feasibility of constructing models of apprentices of a sensorimotor skill that are inspectable, understandable and modifiable by the same apprentices is not obvious. Research has shown that the skill to perform sensorimotor tasks, and other more cognitive control tasks, relies on knowledge that is not easily verbalizable. The same tasks are generally tackled by intelligent computer programs using non-symbolic techniques such as neural networks, genetic algorithms and reinforcement learning. Research using symbolic representations of the knowledge (such as production rules) to accomplish this sort of task have been devoted to acquiring expert knowledge without requiring it to be comprehensible by apprentices of the skill.

The fact that it has been possible to build learner models for the skill of controlling the pole and cart, which can be presented to novices in a meaningful way, speaks for the generality of the participative approach to learner modelling. Future work can be carried out on improving the interface to the current models, for example, by using more comprehensible presentations of their contents, providing better explanation facilities and raising the level of abstraction in the presentation.

The main characteristic of the learner models constructed in this research is their power to predict learner actions. It seems reasonable that the accuracy of their predictions can be improved by making more informed and refined estimates of reaction delays (e.g. reaction delays per rule) and including these estimates in the learner models. Nevertheless, the learner models constructed in this research embody a somewhat simplistic conception of the skill of controlling the pole and cart which departs from the more elaborated approaches described by some of the participants in the studies. Future research can explore constructing learner models based on more structured conceptions of the skill.

The learner modelling method developed in this research produces models with a minimum of assumptions about the domain. Although it has been argued here that this property helps to produce learner models tailored to the individual learner, and not to a predefined conceptuali-

sation of the domain, incorporating domain knowledge in the modelling process can improve the quality of the learner models it produces. In particular, it may permit constructing learner models that are not disposable, as they currently are.

### 8.3.2 Concrete hypotheses and empirical results

This research differs from previous work also in the effort put into testing whether learner's reflection on and awareness of his own knowledge actually took place. It contributes to the answer by deriving a conclusion from the now classical view of the human cognitive architecture as an information-processing system: that learner's reflection should lead to the acquisition of new or the reinforcement of existent conceptual knowledge and meta-knowledge about the domain. From this conclusion, a set of three working hypotheses of detectable changes in learner behaviour consequent to participative learner modelling have been proposed: (i) increased ability to articulate accurate domain knowledge; (ii) increased ability to use domain knowledge in flexible ways, and (iii) a possible initial decrease in performance of the original skill. The results of the set of studies conducted in this research suggest participative learner modelling increases the ability of learners to articulate their knowledge, which can be interpreted as increased knowledge awareness. No improvements in other tasks, more related to knowledge transferability and flexibility, were found.

It can be argued, though, that the *real* educational benefits of participative learner modelling could not show up in the small scale of the studies (of short duration and small sample size), so longer studies involving a greater number of people are two of the obvious directions to follow in future investigations on the effects of participative learner modelling. Other improvements to the design of studies in future research have been suggested in Section 8.1.

Cognitive psychology is a very active field of research, where very little (if anything) can be deemed not subject to dispute. Even theories that generally agreed with the tenets of the information-processing view of human cognition differ in enough details to produce significantly different prediction. A promising direction to extend this research is to incorporate the details of more specific "classical" theories for producing even more concrete hypotheses about the effects of participative learner modelling. Another interesting line of research is to consider "alternative" theories about human cognition, such as those that embrace the tenets of connectionism and situated cognition.

Generalisation and specialisation of the results obtained in this research require additional work to be done using other domains of application, involving both perceptual-motor and cognitive skills. Two important questions to address in this direction are which domains benefit

more from participative learner modelling and why. Future investigations can take advantage of the directions suggested by this research and design more focused, refined and powerful explorations of the terrain.

# Bibliography

- Akhras, F. N. & Self, J. A. (1997). Modelling learning as a process. In du Boulay & Mizoguchi (1997), pp. 418–425.
- Aleven, V. & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help. In Gauthier et al. (2000), pp. 292–303.
- Anderson, J. R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, Mass., USA.
- Anderson, J. R. (1993). *Rules of the Mind*. Lawrence Erlbaum Associates.
- Anderson, J. R., Boyle, C. F., Corbett, A. T. & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42, 7–49.
- Anderson, J. R., Boyle, C. F. & Yost, G. (1985). The geometry tutor. In Joshi (1985), pp. 1–7.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R. & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2), 167–207.
- Anderson, J. R. & Lebiere, C., eds. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum Associates.
- Anderson, J. R. & Singley, M. K. (1993). The identical elements theory of transfer. In Anderson (1993), chap. 9, pp. 183–204.
- Annett, J. (1995). Motor skills. In Mackintosh & Colman (1995), chap. 4, pp. 56–75.
- Ayala, G. & Yano, Y. (1996). Learner models for supporting awareness and collaboration in a CSCL environment. In C. Frasson & G. Gauthier, eds., *Intelligent Tutoring Systems: Third International Conference, ITS'96*, no. 1086 in Lecture Notes in Computer Science, pp. 158–167. Springer-Verlag.

- Baker, M. (1994). A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education*, 5(2), 199–254.
- Barnard, Y. F. & Sandberg, J. A. C. (1996). Self-explanations, do we get them from our students? In Brna et al. (1996), pp. 115–121.
- Barr, A. & Feigenbaum, E. A., eds. (1981). *The Handbook of Artificial Intelligence*, vol. I. Pitman, London.
- Barto, A. G., Sutton, R. S. & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13, 834–846.
- Bechtel, W. & Abrahamsen, A. A. (1991). *Connectionism and the Mind: An Introduction to Parallel Processing in Networks*. Basil Blackwell, Oxford.
- Beck, J., Stern, M. & Woolf, B. P. (1997). Cooperative student models. In du Boulay & Mizoguchi (1997), pp. 127–134.
- Berry, D. C. & Broadbent, D. E. (1984). On the relationship between task-performance and associated verbalizable knowledge. *Quarterly Journal of Experimental Psychology*, 36A(2), 209–231.
- Berry, D. C. & Broadbent, D. E. (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology*, 79(2), 251–272.
- Billings, D. (2000). The international roshambo programming competition. URL <http://www.cs.ualberta.ca/~darse/rsbpc.html>. As on 2 June 2000.
- Bratko, I. (1995). Derivating qualitative control for dynamic systems. In K. Furukawa, D. Michie & S. Muggleton, eds., *Machine Intelligence*, vol. 14, pp. 367–386. Clarendon Press, Oxford.
- Bratko, I., Urbančič, T. & Sammut, C. (1997). Behavioural cloning of control skill. In R. S. Michalski, I. Bratko & M. Kubat, eds., *Machine Learning and Data Mining: Methods and Applications*, chap. 14, pp. 335–351. John Wiley & Sons.
- Brna, P., Paiva, A. & Self, J., eds. (1996). *European Conference on Artificial Intelligence in Education: Proceedings of EuroAIED*. Colibri, Lisbon, Portugal.



- Brna, P., Self, J., Bull, S. & Pain, H. (1999). Negotiated collaborative assessment through collaborative student modelling. In Morales et al. (1999b), pp. 35–42. Also available in (Morales et al., 1999a).
- Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3), 139–159.
- Brown, A. (1987). Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In Weinert & Kluwe (1987), pp. 65–116.
- Brown, J. S. & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2, 155–192.
- Bull, S. (1997a). *Collaborative Student Modelling in Foreign Language Learning*. Ph.D. thesis, University of Edinburgh, Department of Artificial Intelligence.
- Bull, S. (1997b). See yourself write: A simple student model to make students think. In A. Jameson, C. Paris & C. Tasso, eds., *User Modeling: Proceedings of the Sixth International Conference, UM97*, pp. 315–326. Springer Wien New York.
- Bull, S. (in press). Individualised recommendations for learning strategy use. In Gauthier et al. (2000).
- Bull, S., Brna, P. & Pain, H. (1995). Extending the scope of the student model. *User Modeling and User-Adapted Interaction*, 5(1), 45–65.
- Bull, S. & Broady, E. (1997). Spontaneous peer tutoring from sharing student models. In du Boulay & Mizoguchi (1997), pp. 143–150.
- Bull, S. & Ma, Y. (submitted). Raising learner awareness of language learning strategies in situations of limited resources. Submitted for publication.
- Bull, S. & Pain, H. (1995). “Did I say what I think I said, and do you agree with me?” inspecting and questioning the student model. In Greer (1995), pp. 501–508.
- Bull, S. & Shurville, S. (1999). Cooperative writer modelling: Facilitating reader-based writing with scrawl. In Morales et al. (1999b), pp. 1–8. Also available in (Morales et al., 1999a).
- Burton, R. B. & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In Sleeman & Brown (1982), chap. 4, pp. 79–98.

- Byrne, M. D. & Anderson, J. R. (1998). Perception and action. In Anderson & Lebiere (1998), pp. 167–200.
- Camacho, R. (1998). Inducing models of human control skills. In C. Nédellec & C. Rouveirol, eds., *Machine Learning: ECML-98—10th European Conference on Machine Learning*, no. 1398 in Lecture Notes in Artificial Intelligence, pp. 107–118. Springer-Verlag.
- Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer assisted instruction. *IEEE Transactions on Man Machine Systems*, MMS-11(4), 190–202.
- Cheng, P. W. (1985). Restructuring versus automaticity: Alternative accounts of skill acquisition. *Psychological Review*, 92(3), 414–423.
- Clancey, W. J. (1983). GUIDON. *Journal of Computer Based Instruction*, 10(1–2), 8–15.
- Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3(4), 261–283. Also available as a WWW document from <http://www.cs.utexas.edu/users/pclark/software.html>.
- Cohen, W. W. (1995). Fast effective rule induction. In A. Frieditis & S. Russell, eds., *Machine Learning: Proceedings of the Twelfth International Conference*. Morgan Kaufmann. Available as a WWW document from <http://www.research.att.com/~wcohen/ripperd.html>.
- Collins, A. & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold, eds., *Learning Issues for Intelligent Tutoring Systems*, pp. 1–18. Springer-Verlag.
- Conlon, T. (1999). Alternatives to rules for knowledge-based modelling. *Instructional Science*, 27(6), 117–146.
- Conlon, T., Cox, R., Lee, J., McKendree, J. & Stenning, K. (1999). Investigating representational competence in secondary school students. In Lajoie & Vivet (1999), pp. 632–637.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. John Wiley & Sons, 3rd edn.
- Cook, R. & Kay, J. (1994). The justified user model: A viewable, explained user model. In User Modeling Inc. (1994), pp. 145–150.
- Coolican, H. (1994). *Research Methods and Statistics in Psychology*. Hodder & Stoughton, London, 2nd edn.

- Corbett, A. T., Anderson, J. R. & Patterson, E. G. (1990). Student modelling and tutoring flexibility in the lisp intelligent tutoring system. In C. Frasson & G. Gauthier, eds., *Intelligent Tutoring Systems: At the CrossRoad of Artificial Intelligence and Education*, chap. 4, pp. 83–106. Ablex, Norwood, N.J.
- Cox, R. & Brna, P. (1995). Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education*, 6(2/3), 239–302.
- Cratty, B. J. (1973). *Teaching Motor Skills*. Prentice Hall.
- Darnell, E., Guterman, L. & Johnson, T. (1998). Antz. DreamWorks SKG and Pacific Data Images. Film.
- Dasgupta, D. (1998). Evolving neuro-controllers for a dynamic system using structured genetic algorithms. *Applied Intelligence*, 8(2), 113–121.
- Data Protection Registrar (1999). Data protection registrar. WWW document at <http://www.dataprotection.gov.uk>.
- de Buen, P. R., Vadera, S. & Morales, E. F. (1999). A collaborative approach to user modeling within a multi-functional architecture. In Kay (1999), pp. 291–293.
- Diagrams 2000 (2000). *Diagrams 2000*. URL <http://www-cs.hartford.edu/~d2k>.
- Dienes, Z. & Perner, J. (1999). A theory of implicit and explicit knowledge. *Behavioural and Brain Sciences*, 22(735–808).
- Dillenbourg, P. (1992). The computer as a constructorium: Tools for observing one's own learning. In M. Elsom-Cook & R. Moyse, eds., *Knowledge Negotiation*, chap. 8, pp. 185–198. Academic Press.
- Dillenbourg, P. & Self, J. (1992). PEOPLE POWER: A human-computer collaborative learning system. In C. Frasson, Gilles Gauthier & G. McCalla, eds., *Intelligent Tutoring Systems: Second International Conference, ITS'92*, no. 608 in Lecture Notes in Computer Science, pp. 651–660. Montréal, Canada.
- Dimitrova, V., Self, J. & Brna, P. (1999a). The interactive maintenance of open learner models. In Lajoie & Vivet (1999), pp. 405–412.

- Dimitrova, V., Self, J. & Brna, P. (1999b). STyLE-OLM—an interactive diagnosis tool in a terminology learning environment. In Morales et al. (1999b), pp. 25–34. Also available in (Morales et al., 1999a).
- Dimitrova, V., Self, J. A. & Brna, P. (2000). Maintaining a jointly constructed student model. Tech. Rep. 00/5, Computer Based Learning Unit, University of Leeds.
- du Boulay, B., Luckin, R. & del Soldato, T. (1999). The plausibility problem: Human teaching tactics in the ‘hands’ of a machine. In Lajoie & Vivet (1999), pp. 225–232.
- du Boulay, B. & Mizoguchi, R., eds. (1997). *Artificial Intelligence in Education: Knowledge and Media in Learning Systems. Proceedings of the AI-ED 97 World Conference on Artificial Intelligence in Education*. IOS Press.
- Eastwood, E. (1968). Control theory and the engineer. *Proceedings of the Institution of Electrical Engineers*, 115(1), 203–211.
- Elsom-Cook, M. (1993). Student modelling in intelligent tutoring systems. *Artificial Intelligence Review*, 7(3–4), 227–240.
- Ericsson, K. A. & Lehmann, A. C. (1996). Expert and exceptional performance: Evidence of maximal adaptation to task constraints. *Annual Review of Psychology*, 47, 273–305.
- Ericsson, K. A. & Oliver, W. L. (1995). Cognitive skills. In Mackintosh & Colman (1995), chap. 3, pp. 37–55.
- Ericsson, K. A. & Simon, H. A. (1984). *Protocol Analysis: Verbal Reports as Data*. A Bradford book. MIT Press.
- Feynman, R. P., Leighton, R. B. & Sands, M. (1963). *The Feynman Lectures on Physics*. Addison Wesley.
- Finton, D. J. (1994). Controller-less driver for the cart-pole problem. WWW document at <http://www.cs.wisc.edu/~finton/poledriver.html>.
- Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton, ed., *Categories of Human Learning*, pp. 243–285. Academic Press.
- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-development. *American Psychologist*, 34(10), 906–911.

- Flavell, J. H. (1987). Speculations about the nature and development of metacognition. In Weinert & Kluwe (1987), chap. 1, pp. 21–29.
- Fodor, J. A. (1983). *The Modularity of Mind*. MIT Press.
- Gauthier, G., Frasson, C. & VanLehn, K., eds. (2000). *Intelligent Tutoring Systems: Fifth International Conference, ITS'2000*, no. 1839 in Lecture Notes in Computer Science. Springer-Verlag, Montréal, Canada.
- Geddes, B. W. & Stevenson, R. J. (1997). Explicit learning of a dynamic system with a non-salient pattern. *Quarterly Journal of Experimental Psychology*, 50A(4), 742–765.
- Genesereth, M. R. (1982). The role of plans in intelligent teaching systems. In Sleeman & Brown (1982), chap. 7, pp. 137–155.
- Gilmore, D. & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self, ed., *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instruction*, chap. 1, pp. 179–196. Chapman and Hall Computing, London.
- Gluck, K. A., Anderson, J. R. & Douglass, S. A. (2000). Broader bandwidth in student modelling: What if ITS where “Eye”TS. In Gauthier et al. (2000), pp. 504–513.
- Goldstone, R. L. (1998). Perceptual learning. *Annual Review of Psychology*, 49, 585–612.
- Good, I. J. (1965). *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. MIT Press.
- Greer, J. E., ed. (1995). *Artificial Intelligence in Education, 1995: Proceedings of the Seventh World Conference on Artificial Intelligence in Education*. Washington, DC.
- Greer, J. E. & McCalla, G. I., eds. (1994). *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, vol. 125 of NATO ASI Series F: Computer and Systems Sciences. Springer-Verlag.
- Haider, H. & Frensch, P. A. (1996). The role of information reduction in skill acquisition. *Cognitive Psychology*, 30, 304–337.
- Hannafin, M. J. & Land, S. M. (1997). The foundations and assumptions of technology-enhanced student-centred learning environments. *Instructional Science*, 25, 167–202.

- Hayes-Michie, J. (1999). *The Structure of Interactive Decision Skill in Novice and Experts*. Ph.D. thesis, University of Strathclyde, Glasgow, UK. Abstract.
- Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, 28(9), 921–932.
- Hays, W. L. (1994). *Statistics*. Harcourt Brace, 5th edn.
- Hietala, P. & Niemirepo, T. (1997). Collaboration with software agents: What if the learning companion agent makes errors? In du Boulay & Mizoguchi (1997), pp. 159–166.
- Holden, S. & Kay, J. (1999). The scrutable user model and beyond. In Morales et al. (1999b), pp. 51–62. Also available in (Morales et al., 1999a).
- Jameson, A. (1996). Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction*, 5(3–4), 193–251.
- Johnson, W. L. (1990). Understanding and debugging novice programs. *Artificial Intelligence*, 42, 51–97.
- Jonassen, D. H. & Grabowski, B. L. (1993). *Handbook of Individual Differences, Learning, and Instruction*. Lawrence Erlbaum Associates.
- Jones, S. (2000). Animated diagrams: An investigation into the cognitive effects of using animation to illustrate dynamic processes. In Diagrams 2000 (2000). URL <http://www-cs.hartford.edu/~d2k>.
- Joshi, A., ed. (1985). *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. Morgan Kaufman Publishers.
- Joyce, B., Calhoun, E. & Hopkins, D. (1997). *Model of Learning—Tools for Teaching*. Open University Press, Buckingham.
- Karmiloff-Smith, A. (1992). *Beyond Modularity: A Developmental Perspective on Cognitive Science*. MIT Press.
- Karmiloff-Smith, A. (1996). Internal representations and external notations: a developmental perspective. In D. Peterson, ed., *Forms of Representation: an Interdisciplinary Theme for Cognitive Science*, pp. 141–151. Intellect, Exeter.

- Kay, J. (1994a). Lies, damned lies and stereotypes: Pragmatic approximations of users. In User Modeling Inc. (1994), pp. 175–184.
- Kay, J. (1994b). The um toolkit for reusable, long term user models. *User Modeling and User-Adapted Interaction*, 4(3), 149–196.
- Kay, J. (1997). Learner know thyself: Student models to give learners control and responsibility. In Z. R. Z. Halim, T. Ottomann, ed., *Proceedings of the International Conference on Computers in Education, ICCE'97*, pp. 17–24. Kuching.
- Kay, J., ed. (1999). *UM99 User Modeling: Proceedings of the Seventh International Conference*. Springer Wien New York.
- Kay, J. (2000). Accretion representation for scrutable student modelling. In Gauthier et al. (2000), pp. 514–523.
- Kay, J. & Thomas, R. C. (1995). Studying long-term system use. *Communications of the ACM*, 38(7), 61–69.
- Kerr, R. M. & Kibira, D. (1994). Intelligent reactive scheduling by human learning and machine induction. In *IFAC Symposium on Intelligent Manufacturing*. Viena, Austria.
- Koedinger, K. R. & Anderson, J. R. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30–43.
- Kohavi, R. (1996). Machine learning library in C++. WWW document at <http://www.sgi.com/Technology/mlc>. Silicon Graphics, Inc.
- Kohavi, R., Sommerfield, D. & Dougherty, J. (1997). Data mining using MLC++, a machine learning library in C++. *International Journal on Artificial Intelligence Tools*, 6(4), 537–566.
- Kuo, B. C. (1995). *Automatic Control Systems*. Prentice Hall, 7th edn.
- Kurfess, F. J. (1999). Neural networks and structured knowledge: Knowledge representation and reasoning. *Applied Intelligence*, 11(1), 5–13.
- Laird, J. E. & van Lent, M. (2000). Human-level AI's killer application: Interactive computer games. In *The Seventeenth National Conference on Artificial Intelligence*. American Association for Artificial Intelligence. URL <http://ai.eecs.umich.edu/people/laird/papers/AAAI-00.pdf>. Invited talk.

- Lajoie, S. P. & Vivet, M., eds. (1999). *Artificial Intelligence in Education—Open Learning Environments: New Computational Technologies to Support Learning, Exploration, and Collaboration; proceedings of the 9th International Conference on Artificial Intelligence in Education*. IOS Press.
- Langley, P., Ohlsson, S. & Sage, S. (1984). A machine learning approach to student modeling. Technical Report CMU-RI-TR-84-7, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.
- Lebiere, C. & West, R. L. (1999). A dynamic ACT-R model of simple games. In M. Hahn & S. C. Stoness, eds., *Proceedings of the Twenty First Annual Conference of the Cognitive Science Society*, pp. 296–301. Lawrence Erlbaum Associates.
- Lee, F. J. & Anderson, J. R. (1997). Learning to act: Acquisition and optimization of procedural skills. In M. G. Shafto & P. Langley, eds., *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp. 418–423. Lawrence Erlbaum Associates.
- Lee, M. H. (1999). On models, modelling and the distinctive nature of model-based reasoning. *AI Communications*, 12, 127–137.
- Lowe, R. K. (2000). Animation of diagrams: An aid to learning. In *Diagrams 2000* (2000). URL <http://www-cs.hartford.edu/~d2k>.
- Mackintosh, N. J. & Colman, A. M., eds. (1995). *Learning and Skills*. Longman Essential Psychology. Longman.
- Maglio, P. P. (1995). *The Computational Basis of Interactive Skill*. Ph.D. thesis, University of California, San Diego.
- Makarovic, A. (1991). A qualitative way of solving the pole balancing problem. In J. E. Hayes, D. Michie & E. Tyugu, eds., *Machine Intelligence*, vol. 12. Clarendon Press, Oxford.
- McCalla, G. (2000). The fragmentation of culture, learning, teaching and technology: Implications for the artificial intelligence in education research agenda in 2010. *International Journal of Artificial Intelligence in Education*, 11, 177–196.
- Meixler Technologies, Inc. (1999). Meixler technologies, inc. software and systems solutions provider: Inverted pendulum laboratory manual. WWW document at <http://www.meixler-tech.com/manual.htm>.



- Meyer, D. E. & Kieras, D. E. (1997a). A computational theory of executive cognitive processes and multiple-task performance: Part 1. basic mechanisms. *Psychological Review*, 104(1), 3–65.
- Meyer, D. E. & Kieras, D. E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part 2. accounts of psychological refractory-period phenomena. *Psychological Review*, 104(4), 749–791.
- Michie, D. (1993). Knowledge, learning and machine intelligence. In L. Sterling, ed., *Intelligent Systems*. Plenum Press, Oxford.
- Michie, D. (1998). Learning concepts from data. *Expert Systems with Applications*, 15, 193–204.
- Michie, D., Bain, M. & Hayes-Michie, J. (1990). Cognitive models from subcognitive skills. In J. McGhee, M. J. Grimble & P. Mowforth, eds., *Knowledge-Based Systems for Industrial Control*, chap. 5, pp. 71–99. Peter Peregrinus, London.
- Michie, D. & Camacho, R. (1994). Building symbolic representations of intuitive real-time skills from performance data. In K. Furukawa, D. Michie & S. Muggleton, eds., *Machine Intelligence*, vol. 13, pp. 385–418. Clarendon Press, Oxford.
- Michie, D. & Chambers, R. A. (1968). BOXES: An experiment in adaptive control. In E. Dale & D. Michie, eds., *Machine Intelligence*, vol. 2, pp. 137–152. Oliver and Boyd, Edinburgh.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Morales, R. & Pain, H. (1999). Modelling of novices' control skills with machine learning. In Kay (1999), pp. 159–168.
- Morales, R., Pain, H., Bull, S. & Kay, J. (1999a). Open, interactive, and other overt approaches to learner modelling. *AIED Resources*, 10, 1070–1079. WWW document at [http://cbl.leeds.ac.uk/ijaied/abstracts/Vol\\_10/modelling.html](http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_10/modelling.html).
- Morales, R., Pain, H., Bull, S. & Kay, J., eds. (1999b). *Proceedings of the Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling*. AI-ED'99, Le Mans, France. Also available in (Morales et al., 1999a).
- Morales, R., Pain, H. & Conlon, T. (2000). Understandable learner models for a sensorimotor control task. In Gauthier et al. (2000), pp. 222–231.

- Moriarty, D. E. & Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22(1-3), 11-32.
- Murthy, S. K., Kasif, S. & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2, 1-32.
- Narayanan, N. H. & Hegarty, M. (2000). Communicating dynamic behaviors with mixed-mode representations: Comparative studies in two domains. In *Diagrams 2000* (2000). URL <http://www-cs.hartford.edu/~d2k>.
- Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.
- Newell, A. & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, ed., *Cognitive Skills and Their Acquisition*, chap. 1, pp. 1-55. Lawrence Erlbaum Associates.
- Ohlsson, S. (1986). Some principles of intelligent tutoring. *Instructional Science*, pp. 293-326.
- Ohlsson, S. (1992). Artificial instruction: A method for relating learning theory to instructional design. In M. Jones & P. H. Winne, eds., *Adaptive Learning Environments: Foundations and Frontiers*, NATO ASI Series, pp. 55-83. Springer-Verlag.
- Pain, H., Bull, S. & Brna, P. (1996). A student model 'for its own sake'. In Brna et al. (1996), pp. 191-198.
- Paiva, A., Self, J. A. & Hartley, R. (1995). Externalising learner models. In Greer (1995).
- Pineda, L. & Garza, G. (1999). A model for multimodal representation and inference. In R. Paton & I. Neilsen, eds., *Visual Representations and Interpretations*. Springer.
- Poulton, E. C. (1966). Tracking behavior. In E. A. Bilodeau, ed., *Acquisition of Skill*, chap. 8, pp. 361-410. Academic Press.
- Quinlan, R. J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Ramírez Uresti, J. A. (1998). Teaching a learning companion. In G. Ayala, ed., *Proceedings of the Current Trends and Applications of Artificial Intelligence in Education Workshop*, pp. 83-89. ITESM, Mexico City, Mexico.
- Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3), 257-266.

- Reiner, R. (1987). *The princess bride*. Act III Communications, Buttercup Films Ltd. and The Princess Bride Ltd. Film.
- Reiser, B. J., Anderson, J. R. & Farrell, R. G. (1985). Dynamic student modelling in an intelligent tutor for Lisp programming. In Joshi (1985), pp. 8–14.
- Rich, E. (1979). User modelling via stereotypes. *Cognitive Science*, 3, 329–354.
- Roberts, B., Pioch, N. & Ferguson, W. (2000). Verbal coaching during a real-time task. *International Journal of Artificial Intelligence in Education*, 11. To appear.
- Russell, S. & Norvig, P., eds. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- Salvucci, D. D. (1999a). *Mapping Eye Movements to Cognitive Processes*. Ph.D. thesis, Carnegie Mellon University. URL <http://www.cs.cmu.edu/~dario/TH99>. Also available as (Salvucci, 1999b).
- Salvucci, D. D. (1999b). Mapping eye movements to cognitive processes. Tech. Rep. CMU-CS-99-131, Carnegie Mellon University. URL <http://www.cs.cmu.edu/~dario/TH99>.
- Schacter, D. L. (1999). Implicit vs. explicit memory. In R. A. Wilson & F. C. Keil, eds., *The MIT Encyclopedia of the Cognitive Sciences*, pp. 394–395. MIT Press.
- Schlimmer, J. C. & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning*, 1, 317–354.
- Self, J. (1986). The application of machine learning to student modelling. *Instructional Science*, 14(3–4), 327–338.
- Self, J. (1999a). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence in Education*, 10, 350–364.
- Self, J. (1999b). Open sesame?: fifteen variations on the theme of openness in learning environments (aied'99 invited talk). *AIED Resources*, 10, 1020–1029. WWW document at [http://cbl.leeds.ac.uk/ijaied/abstracts/Vol\\_10/self.html](http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_10/self.html).
- Self, J. A. (1988a). Bypassing the intractable problem of student modelling. In *Proceedings of ITS'88*, pp. 18–24. Montréal, Canada.

- Self, J. A. (1988b). Students models: what use are they? In P. Ercoli & R. Lewis, eds., *Artificial Intelligence Tools in Education*, pp. 73–86. North-Holland.
- Self, J. A. (1994a). Dormobile: A vehicle for metacognition. AAI/AI-ED Technical Report 98, Computing Department, Lancaster University, Lancaster, UK.
- Self, J. A. (1994b). Formal approaches to student modelling. In Greer & McCalla (1994), pp. 295–352. Also available as AAI/AI-ED Technical Report 92, Computer Department, Lancaster University, 1994.
- Sison, R. & Shimura, M. (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education*, 9, 128–158.
- Sleeman, D. H. (1982). Inferring (mal) rules from pupil's protocols. In *ECAI-82 : 1982 European Conference on Artificial Intelligence*, pp. 160–164. Orsay, France.
- Sleeman, D. H. & Brown, J. S., eds. (1982). *Intelligent Tutoring Systems*. Academic Press, New York.
- Srinivasan, A. & Camacho, R. (1999). Numerical reasoning with an ilp system capable of lazy evaluation and customised search. *Journal of Logic Programming*, 40(2–3), 185–213.
- Stenning, K. & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logig and implementation. *Cognitive Science*, 19, 97–140.
- Stevens, A., Collins, A. & Goldin, S. E. (1982). Misconceptions in students' understanding. In Sleeman & Brown (1982), chap. 1, pp. 13–24.
- Stillings, N. A., Weisler, S. E., Chase, C. H., Feinstein, M. H., Garfield, J. L. & Rissland, E. L. (1995). *Cognitive Science: An Introduction*. MIT Press, 2nd edn.
- Sun, R., Peterson, T. & Merrill, E. (1996). Bottom-up skill learning in reactive sequential decision tasks. In G. W. Cottrell, ed., *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, pp. 684–689. Lawrence Erlbaum Associates.
- Sun, R., Peterson, T. & Merrill, E. (1999). A hybrid architecture for situated learning of reactive sequential decision making. *Applied Intelligence*, 11(1), 109–127.
- Teasley, S. D. & Roschelle, J. (1993). Constructing a joint problem space: The computer as a tool for sharing knowledge. In S. P. Lajoie & S. J. Derry, eds., *Computers as Cognitive Tools*, chap. 8, pp. 229–258. Lawrence Erlbaum Associates.

- Twidale, M. (1992). Improving error diagnosis using intermediate representations. *Instructional Science*, 20, 359–387.
- Urbančič, T. & Bratko, I. (1994). Reconstructing human skill with machine learning. In A. G. Cohn, ed., *ECAI 94: 11th European Conference on Artificial Intelligence*, pp. 498–502. John Wiley & Sons.
- User Modeling Inc. (1994). *Fourth International Conference on User Modeling: Proceedings of the Conference*. Hyannis, MA.
- van Lent, M. & Laird, J. (1999). Learning hierarchical performance knowledge by observation. In *The Sixteenth International Conference on Machine Learning*.
- VanLehn, K. (1988). Student modelling. In M. C. Polson & J. J. Richardson, eds., *Foundations of Intelligent Tutoring Systems*, chap. 3, pp. 55–78. Lawrence Erlbaum Associates, New Jersey.
- VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, 47, 513–539.
- Vassileva, J., Greer, J. & McCalla, G. (1999a). Openness and disclosure in multi-agent learner models. In Morales et al. (1999b), pp. 43–49. Also available in (Morales et al., 1999a).
- Vassileva, J., Greer, J., McCalla, G. & Deters, R. (1999b). A multi-agent design of a peer-help environment. In Lajoie & Vivet (1999), pp. 38–45.
- Vince, M. A. (1948). The intermittency of control movements and the psychological refractory period. *British Journal of Psychology*, 38, 149–157.
- Watson, J. B. (1913). Psychology as the behaviorist views it. *Psychological Review*, 20, 158–177.
- Webb, G. I., Chiu, B. C. & Kuzmycz, M. (1997). Comparative evaluation of alternative induction engines for Feature Based Modelling. *International Journal of Artificial Intelligence in Education*, 8(2), 97–115.
- Webb, G. I. & Kuzmycz, M. (1996). Feature based modelling: A methodology for producing coherent, consistent, dynamically changing models of agents' competencies. *User Modeling and User-Adapted Interaction*, 5, 117–150.
- Weinert, F. E. & Kluwe, R. H. (1987). *Metacognition, Motivation, and Understanding*. Lawrence Erlbaum Associates.

- Weiss, S. M. & Indurkha, N. (1998). *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann Publishers.
- Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Morgan Kaufmann, California.
- Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, 3(1). Reprinted as a book by Academic Press.
- Winograd, T. (1975). Frame representations and the declarative/procedural controversy. In D. C. Bobrow & A. Collins, eds., *Representation and Understanding: Studies in Cognitive Science*, pp. 185–210. Academic Press.
- Zapata-Rivera, J.-D. & Greer, J. E. (2000). Inspecting and visualizing distributed bayesian student models. In Gauthier et al. (2000), pp. 544–553.

## Appendix A

# Dynamics and control of a pole on a cart

This appendix describes a classical approach in control theory to solve the problem of controlling a pole on a cart: first modelling the system in terms of differential equations and then solving the equations using mathematical techniques. The first step is completed in detail, whereas the second step is only sketched.

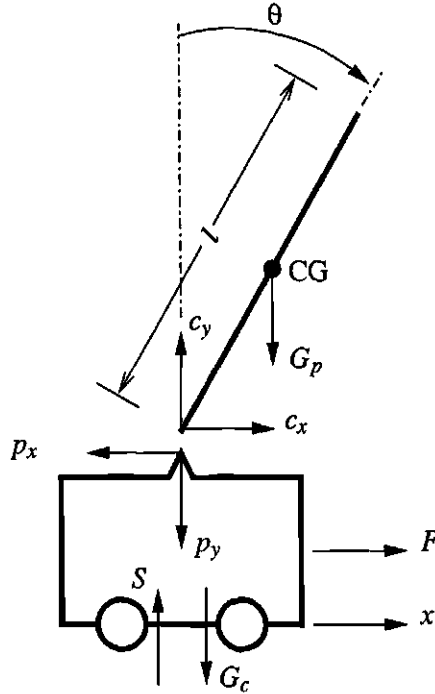
### A.1 Mathematical modelling

The diagram in Figure A.1 shows the relevant forces that produce the movement of a pole and cart device—‘pole and cart’ for short. There are two external forces acting on it: the gravitational force and the force exerted by the controller. The gravitational force affects directly the movement of the pole (and keeps the cart on the rail); the force exerted by the controller affects directly the movement of the cart. The interaction between the two dynamics, due to the fact that the pole is hinged to the cart, is ruled by Newton’s Third Law of Motion, which says that to the force  $P = (p_x, p_y)$  acting from the pole on the cart corresponds another force  $C = (c_x, c_y)$  reacting from the cart on the pole, of equal magnitude but opposite direction. In equations.

$$c_x = -p_x, \tag{A.1}$$

$$c_y = -p_y. \tag{A.2}$$

Newton’s First Law of Motion says the total acceleration of the cart is proportional to the total force applied to it, which is in turn the sum of the external force  $F = (f, 0)$ , the gravitational force  $G_c = (0, -Mg)$ , the force exerted by the pole  $P = (p_x, p_y)$  and the supporting



**Figure A.1:** Free-body diagram of the pole and cart device.

force  $S = (0, s_y)$  from the rail on the cart. The constant of proportionality is given by the inverse of the mass of the cart. In equations:

$$\ddot{x} = \frac{1}{M}(f + p_x), \quad (\text{A.3})$$

$$\ddot{y} = \frac{1}{M}(-Mg + p_y + s_y) = 0. \quad (\text{A.4})$$

The vertical acceleration of the cart is zero because the vertical position of the cart never changes. Besides that, Equation (A.4) gives very little extra information—only that  $s_y$  will be as big as necessary to equilibrate with  $p_y - Mg$ .

In a similar way, Newton's First Law of Motion says the angular acceleration of the pole is proportional to the total rotational force applied to it. In this case, the constant of proportionality is given by the inverse of the moment of inertia—the equivalent to the mass for rotational movement (Feynman et al., 1963, Ch. 18–19). Both rotational forces and moment of inertia can be calculated about any point on the pole, but calculating them about the centre of mass CM is simpler because the gravitational force does not show up. The moment of inertia about the centre of mass is given by the formulae

$$I = \frac{ml^2}{12}. \quad (\text{A.5})$$



The total rotational force, or *torque*, is given by the component of the force  $C$  from the cart on the pole that is perpendicular to the pole times the distance from the centre of mass to the extreme where the force is applied; i.e.

$$\tau = \frac{l}{2} (-c_x \cos \theta + c_y \sin \theta). \quad (\text{A.6})$$

Taking Equations (A.5) and (A.6) together, the resulting equation for the angular acceleration is

$$\ddot{\theta} = \frac{6}{lm} (-c_x \cos \theta + c_y \sin \theta). \quad (\text{A.7})$$

Another useful property of the centre of mass is that, in a sense, it resumes the non-rotational dynamics of the pole; i.e. it accelerates as the sum of the forces on the pole irrespective of what point of the pole each force is applied to. If the position of the centre of mass CM is denote by the coordinates  $(x_m, y_m)$ , the previous statement can be expressed formally as

$$\ddot{x}_m = \frac{1}{m} c_x, \quad (\text{A.8})$$

$$\ddot{y}_m = \frac{1}{m} (c_y - mg). \quad (\text{A.9})$$

Furthermore, since the pole is hinged to the cart at a fixed point, the coordinates for the centre of mass CM can be expressed in terms of the position of the cart and the length and angle of the pole, as follows:

$$x_m = x + \frac{L}{2} \sin \theta, \quad (\text{A.10})$$

$$y_m = y + \frac{L}{2} \cos \theta. \quad (\text{A.11})$$

Double derivation of Equations (A.10) and (A.11), and substitution of the results in Equations (A.8) and (A.9) give, after resolving for  $c_x$  and  $c_y$ :

$$c_x = m\ddot{x} + \frac{ml}{2} (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta), \quad (\text{A.12})$$

$$c_y = mg - \frac{ml}{2} (\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta). \quad (\text{A.13})$$

The substitution of Equations (A.12) and (A.13) for  $c_x$  and  $c_y$ , respectively, in the expression for angular acceleration of the pole gives

$$\ddot{\theta} = \frac{3}{2l} (g \sin \theta - \ddot{x} \cos \theta), \quad (\text{A.14})$$

and the substitution of Equation (A.12) for  $p_x$  in Equation (A.3), through Equation (A.1), gives

$$\ddot{x} = \frac{1}{2(M+m)} (2f - ml (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta)). \quad (\text{A.15})$$

Now, it is only a matter of algebraic manipulations for resolving the last pair of equations for  $\ddot{x}$  and  $\ddot{\theta}$ , in order to produce the final equations that describe the dynamics of the pole and cart, as presented by Bratko (1995):

$$\ddot{x} = \frac{4f + 2ml\dot{\theta}^2 \sin \theta - 1.5mg \sin 2\theta}{4M + 4m - 3m \cos^2 \theta}, \quad (\text{A.16})$$

$$\ddot{\theta} = 6 \frac{(M + m)g \sin \theta - f \cos \theta - 0.5ml\dot{\theta}^2 \sin \theta \cos \theta}{(4M + 4m - 3m \cos^2 \theta)l}. \quad (\text{A.17})$$

These equations are employed to update the state of the pole and cart in the simulator described in Chapter 3.

## A.2 Control

The last equations in the previous section describe the pole and cart as a non-linear system, but they can be simplified (linearised) by assuming that it is possible to keep the pole upright and almost immobile and achieve good control of the cart at the same time. In such a case  $\sin \theta \approx \theta$ ,  $\cos \theta \approx 1$  and  $\dot{\theta}^2 \sin \theta$  is negligible and Equations (A.16) and (A.17) above become

$$\ddot{x} = \frac{4f - 3mg\theta}{4M + m}, \quad (\text{A.18})$$

$$\ddot{\theta} = 6 \frac{(M + m)g\theta - f}{(4M + m)l}. \quad (\text{A.19})$$

This pair of second-order differential equations can in turn be transformed into a system of four first-order differential equations (state equations) which can be analysed using well known mathematical techniques (Kuo, 1995) to produce an expression for the force  $f$  of the form

$$f = ax + b\dot{x} + c\theta + d\dot{\theta}, \quad (\text{A.20})$$

where  $a, b, c, d$ , are constants determined by the mass of the cart and the mass and length of the pole. This equation specifies the force needed for keeping the pole and cart under control given the position and velocity of the cart and the angle and angular velocity of the pole at any moment. Under “bang-bang” conditions (i.e.  $f$  has fixed magnitude; Section 3.1), Equation (A.20) provides the sign of  $f$ .

## **Appendix B**

### **Material of Study 3**

*This appendix contains the material used in Study 3 and the model answers used for evaluating the answers of the participants.*



EMPIRICAL STUDY ON

## Balancing a pole on a cart and understanding a graphical presentation of a strategy to accomplish the task

Rafael Morales\*

### Description

The goal of this study is to evaluate how easy to understand is the graphical presentation of a strategy for controlling a 'pole on a cart' device.

The study consists of four stages. The first stage requires providing basic background information through a small questionnaire. The second stage consists in playing with the pole on a cart for a while. The third stage requires reviewing the graphical presentation of a fictitious strategy for performing the same task. The fourth and final stage consists in answering a small questionnaire about the general interestingness of the study.

---

\*School of Artificial Intelligence, University of Edinburgh. Address: 80 South Bridge, Edinburgh EH1 1HN, Scotland. Phone: +44 (131) 650 2725. Fax: +44 (0131) 650 6516. Email: R.Morales@ed.ac.uk.

## 1 Providing basic information

Please supply the following information about yourself:

1. Sex:

Male [ ]      Female [ ]

2. Age:

3. Which sport(s) do you practice most?

4. Do you like video-games?

Yes [ ]      No [ ]

5. How do you classify yourself at playing video games?

Inexperienced [ ]      Amateur [ ]      Proficient [ ]      Expert [ ]

## 2 Balancing a pole on a cart

Your first task consists in controlling a computer-based simulation of a rigid pole on top of a cart. The pole can fall over the cart either to the left or to the right (see Figure 1).

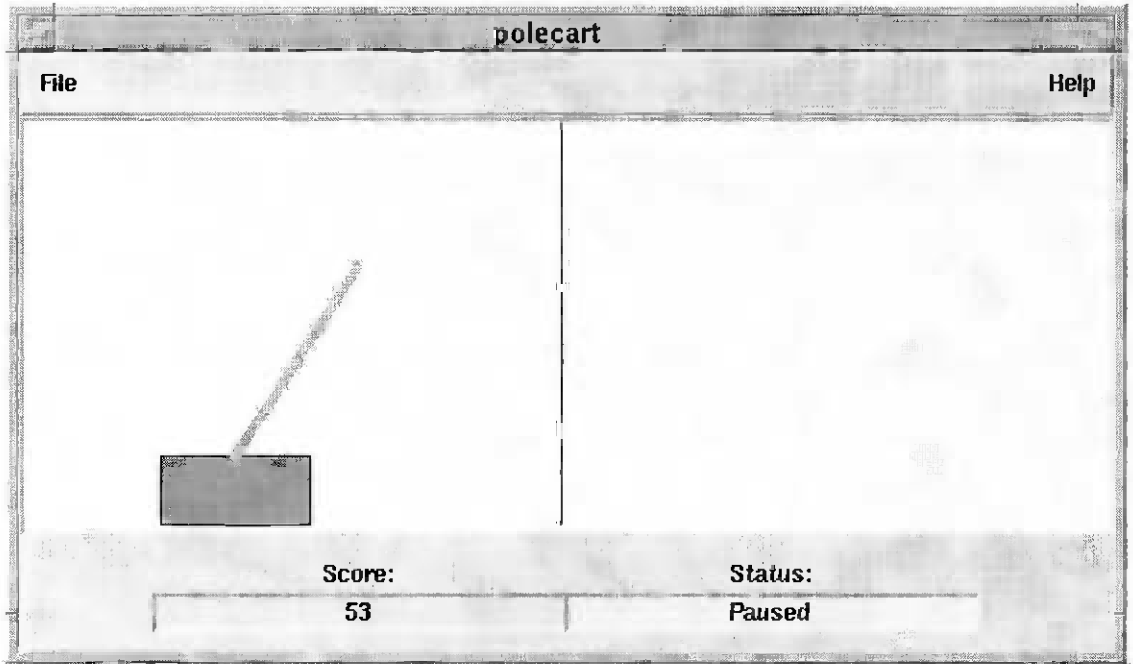


Figure 1: The pole and cart device.

You can control pole and cart only by *pushing the cart* (or not) either to the left or to the right. You will succeed as long as the pole has not reached a horizontal position, and more than half of the cart is inside the window; otherwise you have crashed the thing, and you will need to restart the task. **Please do restart the task every time you have crashed.** Every time you restart the task, the initial position, angle, and velocities of pole and cart will be selected at random.

Input is restricted to pressing arrow keys:  $\uparrow$  to restart the task,  $\leftarrow$  to push the cart to the left, and  $\rightarrow$  to push the cart to the right. Every time you press  $\leftarrow$  or  $\rightarrow$ , and while you keep them pressed, a fixed force will be applied to the cart in the appropriate direction, left or right.

You will have one minute to get familiarised with the program, and then you will be asked to continue on the task for another five minutes. **Please enjoy and do your best!**

### 3 Reviewing a strategy for pole balancing

Your second task consists in reviewing a possible strategy for controlling the pole and cart—not necessarily a good one! The strategy is presented graphically on the screen, in a table-like format (see Figure 2). Every row represents a “rule” for executing an *action*, either of pushing to the left, pushing to the right, or waiting. The action of a rule is executed only if the actual conditions of the pole and cart—their angle, position and velocities—fulfil the rule’s *preconditions*, as explained below.

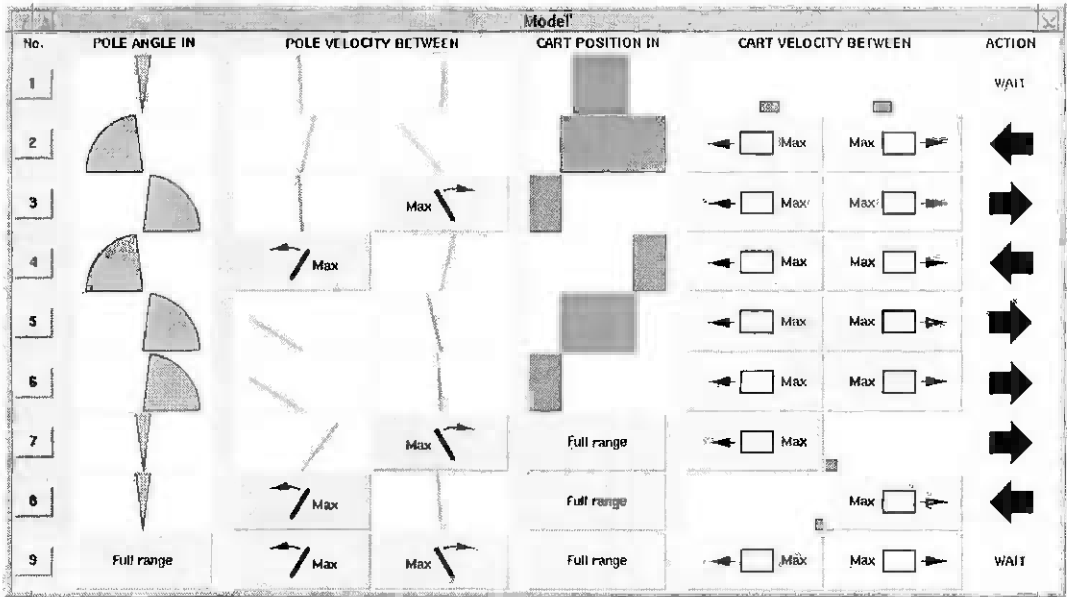


Figure 2: Presentation of a strategy for pole balancing.

The first column in the presentation (No.) contains buttons numbering the rules. The last column (ACTION) contains either the arrow  $\leftarrow$ , which indicates an action of pushing to the left, the arrow  $\rightarrow$ , which indicates an action of pushing to the right, or the word WAIT, which indicates that no action will be executed. The columns from the second to the fifth contain graphical representations of rule preconditions as follows:

- The second column (POLE ANGLE IN) contains drawings of arcs, each one denoting a range of angles for the pole.
- The third column (POLE VELOCITY BETWEEN) contains animations of the pole moving with different velocities. Every pair of animations denote a range of velocities for the pole: the velocities that stand in between the velocities of the animations.
- The fourth column (CART POSITION IN) contains drawings of boxes that represent ranges of positions for the cart. The space in which each box is drawn represents the whole window of the animation.



- The fifth column (CART VELOCITY BETWEEN) contains animations of the cart moving with different velocities. Every pair of animations denote a range of velocities for the cart: the velocities that stand in between the velocities of the animations.

To follow this strategy for balancing the pole on the cart means selecting a rule, and executing its action, if and only if the angle of the pole is in the range of angles represented in the rule's row under POLE ANGLE IN, the velocity of the pole is in the range of velocities represented in the rule's row under POLE VELOCITY BETWEEN, the position of the cart is in the range of positions represented in the rule's row under CART POSITION IN, and the velocity of the cart is in the range of velocities represented in the rule's row under CART VELOCITY BETWEEN.

Please take into account that velocities, are composed of magnitude and direction. The convention used to represent velocities is that they go from *fast-left*—a kind of “negative” velocity—to *fast-right*—a kind of “positive” velocity—with *slow-left*, *zero* and *slow-right* in the middle. For example:

**Example # 1** Two animations representing the pole moving slowly to the left and quickly to the right, respectively, denote pole velocities that are either very slow to the left—not faster than the velocity of the first animation—or not very fast to the right—not faster than the velocity of the second animation.

**Example # 2** Two animations representing the cart moving slowly and quickly to the right, respectively, denote cart velocities to the right that are both faster than the velocity of the first animation and slower than the velocity of the second animation.

The graphical presentation of the strategy contains also textual and iconic representations for full ranges of angles and positions, and maximum speeds. They are introduced for improving the clarity of the presentation and diminishing the overwhelming effect of too many fast animations.

A small window is also displayed on the screen (see Figure 3) which contains two active buttons, Explain and Exit. You can select Explain and then press the button in which a rule's ordinal number is displayed to see a short demonstration of the performance of the model with focus on that rule. The action of that rule, and of any other rule executed in the demonstration, will be highlighted.

**Please study the strategy carefully, and then respond to the questionnaire. Please *DO NOT* press the Exit button until you have finished.**

### 3.1 Questionnaire

1. Which rules would result in *pushing to the left actions* (as indicated by the appropriate arrow in the ACTION column)?

Rule numbers: 2,4,8.

2. Which rules would result in *pushing to the right actions*?

Rule numbers: 3,5,6,7.

3. Explain in your own words the precondition on the *cart position* of Rule 7 (indicated in the CART POSITION IN column).

The cart can be in any position.



Figure 3: Palette of operations for reviewing strategies.

4. Explain in your own words the precondition on the *cart position* of Rule 2.  
The cart is not in the “far left” side of the window.
5. Explain in your own words the precondition on the *pole angle* of Rule 3 (indicated in the POLE ANGLE IN column).  
The pole is tilted to the right more than a little.
6. Explain in your own words the precondition on the *pole velocity* of Rule 9 (indicated in the columns headed POLE VELOCITY BETWEEN).  
Any velocity.
7. Explain in your own words the precondition on the *pole velocity* of Rule 8.  
The pole is falling to the left more than slowly.
8. Explain in your own words the precondition on the *pole velocity* of Rule 5.  
The pole is either falling to the left with a moderate velocity, or it is falling to the right with an up to moderately fast velocity.
9. Explain in your own words the precondition on the *cart velocity* of Rule 7 (indicated in the columns headed CART VELOCITY BETWEEN).  
Any velocity to the left, or a moderate velocity to the right.
10. Explain in your own words the precondition on the *cart velocity* of Rule 1.  
Slowly in either direction.
11. Explain in your own words the preconditions and outcome of Rule 3.

If the cart is in the farthest left side, and the pole is tilted to the right more than a bit and either falling in that direction or recovering slowly, then push to the right.

12. Explain in your own words the preconditions and outcome of Rule 6.

If the cart is in the farthest left side, and the pole is tilted to the right more than a bit but it is recovering to the left from slowly to moderately, then push to the right. NOTE: THIS RULE COMPLEMENTS RULE 3.

13. Explain in your own words the preconditions and outcome of Rule 8.

If the pole angle is very small but the pole is falling to the left more than slowly, and the cart is not moving to the left very fast, then push to the left.

14. Try to provide a description, in your own words, of the overall strategy defined by the rule set.

The strategy is a sort of “natural” one, with a touch of laziness. That means doing nothing if both pole and cart are centred and moving slowly, pushing in the direction of pole’s falling if it accelerates, but not if the cart is moving fast in that direction, and in general trying to revert to a centred position as long as the cart is not too close to an edge (with the pole falling more than slowly towards the edge) or the pole is not falling too quickly.

## 4 Overall evaluation

Finally, I would like you to answer the following small questionnaire:

1. Please provide an evaluation of the first task, **Balancing a pole on a cart**.

Tiresome [ ]    Dull [ ]    Interesting [ ]    Exciting [ ]

Comments:

2. Please provide an evaluation of the second task, **Reviewing a strategy for pole balancing**.

Tiresome [ ]    Dull [ ]    Interesting [ ]    Exciting [ ]

Comments:

3. Do you have any further suggestions for improving the presentation of a strategy for balancing the pole on the cart?

Yes [ ]    No [ ]

Suggestions:

4. Finally, please write down any comments you might have about this study.

Comments:

**THANK YOU!**



## **Appendix C**

### **Material of Experiment 1**



# Graphical Interface to User Models

Rafael Morales\*

September 1, 2000

## 1 Introduction

A *user model* for the pole and cart program is the system's interpretation of the way you try to keep the pole and cart under control. It consists of a set of statements that describes the *strategy* the system believes you follow in carrying out the task, and *data* gathered from your interaction with the pole and cart that justify the system's beliefs (Section 2.1, *The meaning of beliefs*). Both strategy and data are presented in a graphical way to make them easier for you to understand.

Facilities exist for editing a user model—if the system is not operating in read-only mode (Section 4.1, *Read-only mode*)—but data cannot be edited, since it is gathered from your behaviour when commanding the pole and cart. On the other hand, data is always available, so that you can evaluate any changes you make to the strategy.

This document describes first the presentation of the user model (Section 2, *Model presentation*), then the additional information that can be displayed about the model (Section 3, *Model justification, verbalisation and execution*), and finally the facilities provided for its editing (Section 4, *Model editing*).

## 2 Model presentation

The system's model of your way of controlling the pole and cart is presented graphically on the screen in a table-like format (see example in

---

\*School of Artificial Intelligence, University of Edinburgh. Address: 80 South Bridge, Edinburgh EH1 1HN, Scotland, UK. Phone: +44 (0)131 650 2725. Fax: +44 (0)131 650 6516. Email: R.Morales@ed.ac.uk.

Figure 1). Every row presents a "rule" that links a set of states of the pole and cart, described in terms of *conditions* on their angle, position and velocities, with the particular *action* the system believes you will perform if the conditions are fulfilled. Possible actions are either pushing to the left or to the right, or simply waiting for a change of conditions.

**Important:** A rule should not be interpreted as a statement of what you should do when the state of the pole and cart meets certain conditions, but as an statement of *what you generally do* when that conditions are met.

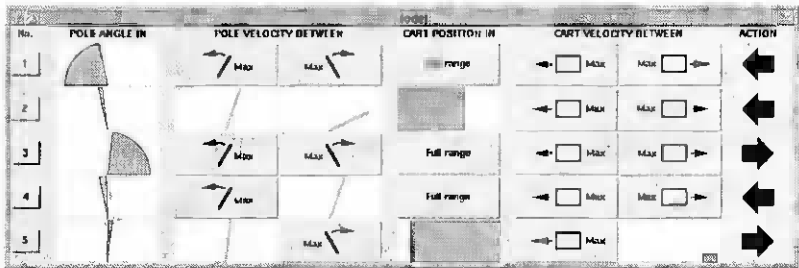


Figure 1: Presentation of a model of user's controlling of the pole and cart.

The first column in the presentation of the user model, headed No., contains buttons numbering the rules. The last column, headed ACTION, contains either an arrow indicating an action of pushing to the left (◀) or pushing to the right (▶), or the word WAIT indicating that no action is executed. The remaining columns, from the second to the fifth, contain graphical representations of sets of conditions on the state of the pole and cart, as follows:





- The second column, headed POLE ANGLE IN, contains drawings of arcs, each one denoting a range of angles for the pole.
- The third column, headed POLE VELOCITY BETWEEN, contains animations of the pole moving with different velocities. Every pair of animations denotes a range of velocities for the pole; namely, the velocities in between the velocities of the animations.
- The fourth column, headed CART POSITION IN, contains drawings of boxes that represent ranges of positions for the cart. The space in which a box is drawn represents the whole window in which the animation of the pole and cart is presented.



- The fifth column, headed CART VELOCITY BETWEEN, contains animations of the cart moving with different velocities. Every pair of animations denotes a range of velocities for the cart; namely, the velocities in between the velocities of the animations.

Velocities consist of both magnitude and direction. The convention used to present pole velocities is that they go from very fast anticlockwise to very fast clockwise. The convention to present cart velocities is that they go from very fast to the left to very fast to the right.

The graphical presentation of the user model may contain the string FULL RANGE, under POLE ANGLE IN and CART POSITION IN, denoting full ranges of angles and positions, respectively. It can also contain the following iconic representations:

-  for maximum speed anticlockwise,
-  for maximum speed clockwise,
-  for maximum speed to the left, and
-  for maximum speed to the right.

They are introduced to improve the clarity of the presentation and to diminish the overwhelming effect of seeing too many fast animations.

## 2.1 The meaning of beliefs

To say that the system believes that you follow the strategy presented on the screen means that the system believes that, at every time during your interaction with the simulation of the pole and cart, either

- you select a rule, and execute its action, whenever the angle and velocity of the pole and the position and velocity of the cart are in the ranges that constitute the rule's conditions; or
- you wait for a change in the state of the pole and cart, whenever the state of the pole and cart does not completely satisfy any rule's conditions.

The system beliefs are justified by data gathered during your interaction with the simulation of the pole and cart.

### 3 Model justification, verbalisation and execution

Beside the window presenting the user model there is another window displayed on the screen (Figure 2) which contains buttons for selecting operations to apply to rules (see also Section 4.1, *Read-only mode*, and Section 4.2, *Operations on rules*). To apply an operation to a rule, you first press the operation's button and then the rule's button—the button on which the ordinal number of the rule is displayed, in the first column of the presentation of the model.



Figure 2: Window with operations to apply to rules.

#### 3.1 Justification

You can select *Justify* and then a rule from the user model to see a graphical summary of data in favour and against the rule (see example in Figure 3).

#### 3.2 Verbalisation

You can select *Verbalise* and then a rule from the user model to see a textual description of that rule (see example in Figure 4).

**Important:** You should regard the textual description of a rule as a complement to, but not a substitute for the graphical description.

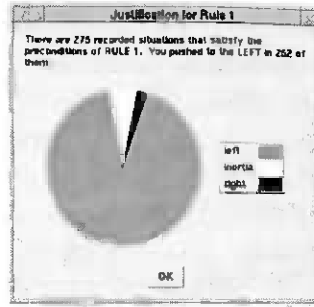


Figure 3: Summary of data in favour and against a rule.

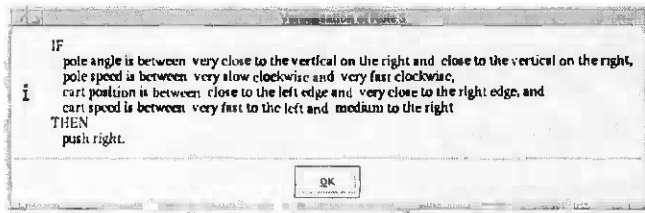


Figure 4: Textual description of a rule.

### 3.3 Runs

If you select Verbalise and a rule from the user model, you will see a short demonstration of the performance of the model at commanding the pole and cart. The initial state of the pole and cart will be selected to match the conditions of the rule selected, and then the strategy will be followed as the state of the pole and cart changes over time. Each action executed will be highlighted in turn and presented as an arrow on the simulation of the pole and cart.

## 4 Model editing

As it was said before (Section 1, *Introduction*), facilities are provided for editing the user model. Select the green button labelled Commit on the operations' window (Figure 2) to make all modifications to the user model permanent and terminate the session. Select the red button labelled Cancel to abort a session without saving any changes to the model.

**Important:** In the original user model, as generated by the

system based on your behaviour, all rules are defined such that every state of the pole and cart matches the conditions of one, and only one, rule. That constraint is not enforced when you edit the model, though, and it is possible for a state of the pole and cart to match the conditions of more than one rule. In such a case *the order of the rules matters*, and the first match determines the action to be selected for that state.

## 4.1 Read-only mode

Commit is disabled and Exit is substituted for Cancel when the graphical interface to the user model is running in read-only mode. All editing operations described in the remaining sections are disabled in read-only mode.

## 4.2 Operations on rules

To apply an operation to a rule, first select the operation, by pressing the corresponding button in the operations' window (Figure 2), and then select the rule, by pressing the button on which the ordinal number of the rule is displayed, in the first column of the window that presents the user model (Figure 1).

### 4.2.1 New

New rules can be added to the user model, either immediately above or below an existing rule. To do that, first select the New above or New below buttons, respectively, and then select the appropriate rule.

### 4.2.2 Cut and copy

An existing rule can be removed from its actual position in the user model by first selecting the Cut button and then selecting the rule. You can also simply make a copy of an existing rule, without removing it from the model, by first selecting the Copy button. Only a copy of the last rule cut or copied is kept by the system.

### 4.2.3 Paste

The copy of the last rule cut or copied from the user model can be copied back into the model, either immediately above or below an existing rule.

To do that, first select the Paste above or Paste below buttons, respectively, and then select the appropriate rule. The operation can be repeated to produce multiple copies of a single rule. The copies can be modified later on using the facilities for editing conditions and actions described in the following section.

### **4.3 Operations on conditions and actions**

A number of facilities are available for editing conditions and actions of rules. In general, the iconic representations of full ranges and maximum velocities (Section 2, *Model presentation*) can be toggled on and off by simultaneously pressing the Control key and the left button on the mouse while the mouse pointer is on an icon, drawing or animation.

#### **4.3.1 Editing of angle and position**

You can modify the conditions on pole angle and cart position, presented in the columns headed POLE ANGLE IN and CART POSITION IN, respectively, by first moving the mouse pointer close to a border of the corresponding arc or box, and then *dragging* the border—i.e. pressing the left button on the mouse and move the mouse while holding the button down—until the border is at the desired position.

#### **4.3.2 Editing of velocities**

Every time the mouse pointer enters the area containing an animation in the columns headed POLE VELOCITY BETWEEN and CART VELOCITY BETWEEN a scale with two sliders will appear: a fixed top slider coloured in red, and a movable bottom slider coloured in green. You can modify the conditions on the car velocity and the pole angular velocity by moving the mouse pointer over the corresponding bottom (green) slider, and dragging it. The top slider on the scale stands for the velocity represented by the companion animation, and hence you cannot move the bottom slider beyond the position of the top slider.

#### **4.3.3 Editing of actions**

You can modify the action of a rule by first moving the mouse pointer over the arrow or word indicating the action, and then pressing the left button on the mouse to change the action to pushing to the left, the right

button to change it to pushing to the right, and the middle button to change it to wait.

# Graphical Interface to Strategies for Controlling the Pole and Cart

Rafael Morales\*

September 1, 2000

## 1 Introduction

A *strategy* for controlling the pole and cart is a set of statements that describes a way of carrying out the task. It is presented in a graphical way to make it easier to understand. Facilities are provided for editing a strategy, when the system is not operating in read-only mode (Section 4.1, *Read-only mode*).

This document describes first the presentation of the strategy (Section 2, *Strategy presentation*), then the additional information that can be displayed about the strategy (Section 3, *Strategy verbalisation and execution*), and finally the facilities provided for its editing (Section 4, *Strategy editing*).

## 2 Strategy presentation

A strategy for controlling the pole and cart is presented graphically on the screen in a table-like format (Figure 1). Every row presents a “rule” that links a set of states of the pole and cart, described in terms of *conditions* on their angle, position and velocities, with a particular *action*. Possible actions are either pushing to the left or to the right, or simply waiting for a change of conditions.

---

\*School of Artificial Intelligence, University of Edinburgh. **Address:** 80 South Bridge, Edinburgh EH1 1HN, Scotland, UK. **Phone:** +44 (0)131 650 2725. **Fax:** +44 (0)131 650 6516. **Email:** R.Morales@ed.ac.uk.

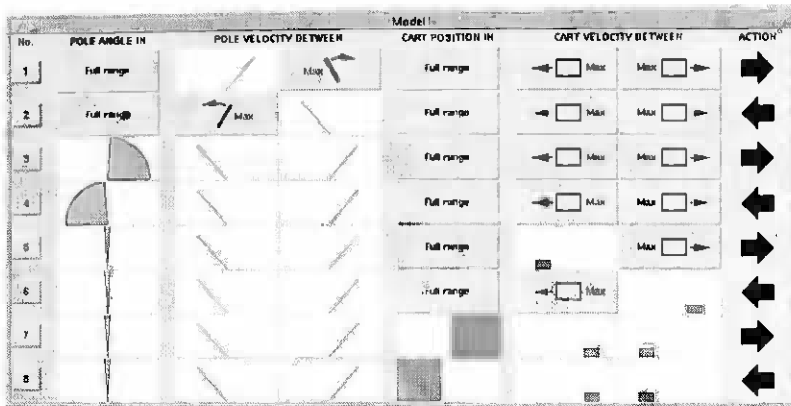


Figure 1: Presentation of a strategy for controlling the pole and cart.

The first column in the presentation of a strategy, headed No., contains buttons numbering the rules. The last column, headed ACTION, contains either an arrow indicating an action of pushing to the left (←) or pushing to the right (→), or the word WAIT indicating that no action is executed. The remaining columns, from the second to the fifth, contain graphical representations of sets of conditions on the state of the pole and cart, as follows:





- The second column, headed POLE ANGLE IN, contains drawings of arcs, each one denoting a range of angles for the pole.
- The third column, headed POLE VELOCITY BETWEEN, contains animations of the pole moving with different velocities. Every pair of animations denotes a range of velocities for the pole; namely, the velocities in between the velocities of the animations.
- The fourth column, headed CART POSITION IN, contains drawings of boxes that represent ranges of positions for the cart. The space in which a box is drawn represents the whole window in which the animation of the pole and cart is presented.
- The fifth column, headed CART VELOCITY BETWEEN, contains animations of the cart moving with different velocities. Every pair of animations denotes a range of velocities for the cart; namely, the velocities in between the velocities of the animations.

Velocities consist of both magnitude and direction. The convention used to present pole velocities is that they go from very fast anticlockwise to



very fast clockwise. The convention to present cart velocities is that they go from very fast to the left to very fast to the right.

The graphical presentation of the strategy may contain the string FULL RANGE, under POLE ANGLE IN and CART POSITION IN, denoting full ranges of angles and positions, respectively. It can also contain the following iconic representations:

-  for maximum speed anticlockwise,
-  for maximum speed clockwise,
-  for maximum speed to the left, and
-  for maximum speed to the right.

They are introduced to improve the clarity of the presentation and to diminish the overwhelming effect of seeing too many fast animations.

### 3 Strategy verbalisation and execution

Beside the window presenting the strategy there is another window displayed on the screen (Figure 2) which contains buttons for selecting operations to apply to rules (see also Section 4.1, *Read-only mode*, and Section 4.2, *Operations on rules*). To apply an operation to a rule, you first press the operation's button and then the rule's button—the button on which the ordinal number of the rule is displayed, in the first column of the presentation of the strategy.

#### 3.1 Verbalisation

You can select Verbalise and then a rule from the strategy to see a textual description of that rule (see example in Figure 3).

**Important:** You should regard the textual description of a rule as a complement to, but not a substitute for the graphical description.

#### 3.2 Runs

If you select Verbalise and a rule from the strategy, you will see a short demonstration of the performance of the strategy at commanding the



Figure 2: Window with operations to apply to rules.

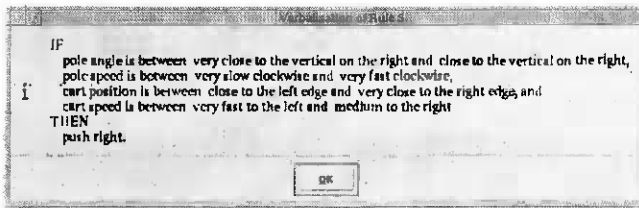


Figure 3: Textual description of a rule.

pole and cart. The initial state of the pole and cart will be selected to match the conditions of the rule selected, and then the strategy will be followed as the state of the pole and cart changes over time. Each action executed will be highlighted in turn and presented as an arrow on the simulation of the pole and cart.

## 4 Strategy editing

As it was said before (Section 1, *Introduction*), facilities are provided for editing the strategy. Select the green button labelled Commit on the operations' window (Figure 2) to make all modifications to the strategy permanent and terminate the session. Select the red button labelled Cancel to abort a session without saving any changes to the strategy.

**Important:** In predefined strategies all rules are defined such that every state of the pole and cart matches the conditions of

one, and only one, rule. That constraint is not enforced when you edit a strategy, though, and it is possible for a state of the pole and cart to match the conditions of more than one rule. In such a case *the order of the rules matters*, and the first match determines the action to be selected for that state.

## 4.1 Read-only mode

Commit is disabled and Exit is substituted for Cancel when the graphical interface to the strategy is running in read-only mode. All editing operations described in the remaining sections are disabled in read-only mode.

## 4.2 Operations on rules

To apply an operation to a rule, first select the operation, by pressing the corresponding button in the operations' window (Figure 2), and then select the rule, by pressing the button on which the ordinal number of the rule is displayed, in the first column of the window that presents the strategy (Figure 1).

### 4.2.1 *New*

New rules can be added to the strategy, either immediately above or below an existing rule. To do that, first select the New above or New below buttons, respectively, and then select the appropriate rule.

### 4.2.2 *Cut and copy*

An existing rule can be removed from its actual position in the strategy by first selecting the Cut button and then selecting the rule. You can also simply make a copy of an existing rule, without removing it from the strategy, by first selecting the Copy button instead. Only a copy of the last rule cut or copied is kept by the system.

### 4.2.3 *Paste*

The copy of the last rule cut or copied from the strategy can be copied back into the strategy, either immediately above or below an existing rule. To do that, first select the Paste above or Paste below buttons, respectively, and then select the appropriate rule. The operation can be

repeated to produce multiple copies of a single rule. The copies can be modified later on using the facilities for editing conditions and actions described in the following section.

### **4.3 Operations on conditions and actions**

A number of facilities are available for editing conditions and actions of rules. In general, the iconic representations of full ranges and maximum velocities (Section 2, *Strategy presentation*) can be toggled on and off by simultaneously pressing the Control key and the left button on the mouse while the mouse pointer is on an icon, drawing or animation.

#### **4.3.1 Editing of angle and position**

You can modify the conditions on pole angle and cart position, presented in the columns headed POLE ANGLE IN and CART POSITION IN, respectively, by first moving the mouse pointer close to a border of the corresponding arc or box, and then *dragging* the border—i.e. pressing the left button on the mouse and move the mouse while holding the button down—until the border is at the desired position.

#### **4.3.2 Editing of velocities**

Every time the mouse pointer enters the area containing an animation in the columns headed POLE VELOCITY BETWEEN and CART VELOCITY BETWEEN a scale with two sliders will appear: a fixed top slider coloured in red, and a movable bottom slider coloured in green. You can modify the conditions on the car velocity and the pole angular velocity by moving the mouse pointer over the corresponding bottom (green) slider, and dragging it. The top slider on the scale stands for the velocity represented by the companion animation, and hence you cannot move the bottom slider beyond the position of the top slider.

#### **4.3.3 Editing of actions**

You can modify the action of a rule by first moving the mouse pointer over the arrow or word indicating the action, and then pressing the left button on the mouse to change the action to pushing to the left, the right button to change it to pushing to the right, and the middle button to change it to wait.

# Experiment material

Rafael Morales\*

September 1, 2000

## Description

Thank you very much for agreeing to participate in this experiment! We aim to test how different tasks in the same domain support the development of different types of knowledge. You will be presented with a series of tasks related to the domain of balancing a pole (or inverted pendulum) hinged to a cart (small vehicle). Some tasks will involve interacting with computer programs and some others will consist in answering short questionnaires. The estimated time to complete the experiment is of about an hour.

## 1 Providing basic information

Firstly, please supply the following information about yourself:

1. Sex:

Male [ ]      Female [ ]

2. Age:

3. Which sport(s), if any, do you practice most?

4. Do you like video-games?

Yes [ ]      No [ ]

5. How do you classify yourself at playing video games?

Beginner [ ]      Amateur [ ]      Proficient [ ]      Expert [ ]

---

\*School of Artificial Intelligence, University of Edinburgh. **Address:** 80 South Bridge, Edinburgh EH1 1HN, Scotland, UK. **Phone:** +44 (0)131 650 2725. **Fax:** +44 (0)131 650 6516. **Email:** R.Morales@ed.ac.uk.

## 2 Balancing a pole on a cart

This part of the experiment consists in you trying to control a computer-based simulation of a rigid pole hinged to a cart (Figure 1). The pole can fall over the cart either to the left or to the right.

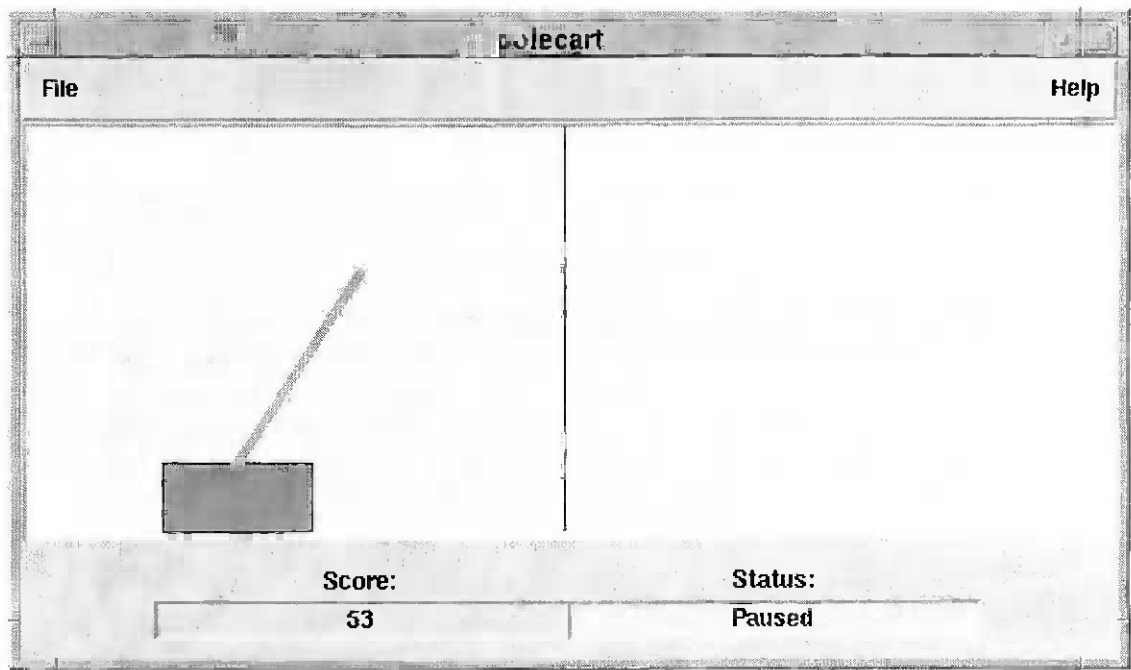


Figure 1: The pole and cart device.

You can control the pole and cart only by pushing the cart, either to the left or to the right. You succeed as long as the pole does not reach a horizontal position and its joint to the cart is inside the window. Otherwise you have crashed the thing, and you will need to restart the task. **Please do restart the task every time you crash.**

Input is restricted to pressing arrow keys:  $\uparrow$  to restart the task,  $\leftarrow$  to push the cart to the left and  $\rightarrow$  to push the cart to the right. A force of fixed magnitude will be applied to the cart in the appropriate direction for as long as you hold the key down.

You have about one minute to become familiar with the program. After that, you will be asked to continue on the task for another five minutes. **Please enjoy and do your best!**

### 3 Inspecting a model of your strategy

You were given additional printed material describing the graphical interface you see now on the screen, which presents the system's interpretation of your behaviour when controlling the pole and cart. In other words, it describes what the system believes *your strategy* is for controlling the pole and cart.

**YOU HAVE GOT 20 MINUTES TO COMPLETE THIS STAGE OF THE EXPERIMENT, including Section 3.1. Please study the model carefully, and then answer the questionnaire below. DO NOT press the Exit button until you have finished.**

#### *Questionnaire*

1. Which rules indicate an action of *pushing to the left*?  
Rule number(s):
2. Which rules indicate an action of *pushing to the right*?  
Rule number(s):
3. Which rules (if any) does the system believe you would select if the pole is halfway down and falling fairly quickly to the right, at the same time as the cart is halfway between the window's centre and left border and it is moving towards the left with moderate speed?  
Rule number(s):
4. Which rules (if any) does the system believe you would select if the pole is almost vertical but falling very slowly to the right, at the same time as the cart is quickly moving towards, and it is very close to, the right border?  
Rule number(s):
5. Which rules (if any) does the system believe you would select if the pole is almost vertical but falling very slowly to the left, at the same time as the cart is moving slowly towards, and it is very close to, the centre from the right?  
Rule number(s):
6. How accurate do you think the system's model of your strategy for controlling the pole and cart is?  
Very accurate [ ]   Mostly accurate [ ]   Mostly inaccurate [ ]   Quite inaccurate [ ]

7. What changes (if any) do you think need to be done in order to improve the accuracy of the model?

### *3.1 Tailoring the model*

Now you have the opportunity to modify the system's model for it to be a more proper description of *your strategy* for controlling the pole and cart. A number of facilities are now available for editing the model, as described in the printed material given to you before.

**Please make any changes to the model for it to better represent your strategy for controlling the pole and cart. Press Commit only after you are satisfied with your model. Press Cancel only if you have made a mistake and want to restart editing the model.**



## 4 A good strategy

You were given additional printed material describing the graphical interface you see now on the screen, which presents what the system believes is the strategy of a skilled user for controlling the pole and cart.

**YOU HAVE GOT 20 MINUTES TO COMPLETE THIS STAGE OF THE EXPERIMENT, including Section 3.1. Please study the strategy carefully, and then answer the questionnaire below. DO NOT press the Exit button until you have finished.**

### *Questionnaire*

8. Which rules indicate an action of *pushing to the left*?

Rule number(s):

9. Which rules indicate an action of *pushing to the right*?

Rule number(s):

10. Which rules (if any) does the system believe the user would select if the pole is halfway down and falling fairly quickly to the right, at the same time as the cart is halfway between the window's centre and left border and it is moving towards the left with moderate speed?

Rule number(s):

11. Which rules (if any) does the system believe the user would select if the pole is almost vertical but falling very slowly to the right, at the same time as the cart is quickly moving towards, and it is very close to, the right border?

Rule number(s):

12. Which rules (if any) does the system believe the user would select if the pole is almost vertical but falling very slowly to the left, at the same time as the cart is moving slowly towards, and it is very close to, the centre from the right?

Rule number(s):

13. How good do you think the strategy is?

Very good [ ]   Good [ ]   Bad [ ]   Very bad [ ]

14. What changes (if any) do you think need to be done in order to improve the strategy?

#### *4.1 Adjusting the strategy*

Now you have the opportunity to modify the system's suggested strategy for it to be better for controlling the pole and cart. A number of facilities are now available for editing the strategy, as described in the printed material given to you before.

**Please make any changes you wish to the strategy. Press Commit only after you are satisfied with your modifications. Press Cancel only if you have made a mistake and want to restart editing the strategy.**

## 5 Improving your skills

Now you have the opportunity to improve your skills for controlling the pole and cart. You have twenty minutes to practice, explore and get quite familiar with the task. Your goal should be to get as good and confident at the task as possible, given the time constraints.



# 6    Playing again

It is time to play again! This time no familiarisation period is set, so you have got a total of five minutes to do your best. **Please do it and have fun!**

## 7 Strategy description

Please answer the following questionnaire, making your responses as detailed as possible.

### *Questionnaire*

15. Try to give an overall description of any strategy you follow for keeping the cart inside the window and avoiding the pole falling over the cart.

16. Describe your main goal when playing with the pole and cart.

17. Do you decompose the goal described before in a number of subgoals? If so, please try to describe them below. [For example, the overall goal of going from one place to another by plane could be decomposed in three subgoals: (1) to put the plane in the air; (2) to reach the destination; and (3) to put the plane back safely on the ground. The subgoal of reaching the destination could in turn be decomposed into two subgoals: (a) keeping the plane at cruise altitude and (b) keeping the plane pointing towards the destination]

18. Under which conditions of the pole and cart do you execute an action of pushing to the right?

19. Under which conditions of the pole and cart do you prefer to wait for a change of conditions?



20. Under which conditions of the pole and cart do you execute an action of pushing to the left?

21. Are there any conditions of the pole and cart you could categorise as the pole and cart being out of your control? Please try to describe them, if any.

22. Rank the following properties of the pole and cart according to their relative importance for controlling purposes. Give them numbers from 1 (most important) to 4 (less important). Assign the same number to all properties you regard as equally important.

Cart position [ ]      Cart velocity [ ]      Pole angle [ ]      Pole velocity [ ]

# 8 Knowledge usage

Please answer the following questionnaire.

## Questionnaire

23. Describe a set of conditions of the pole and cart that you consider quite “safe”, in the sense of being conditions in which you reckon you are in control and you would not push in either direction for a while.

A sequence of ten short animations of the pole and cart will be displayed on the screen. Each animation will end with the pole and cart in different conditions. Please grade the final conditions of each animation using a scale from 1 to 7, with 1 standing for the pole and cart being in conditions totally under control and 7 standing for conditions completely out of control.

24. Under control    1    2    3    4    5    6    7    Out of control

Reason(s):

25. Under control    1    2    3    4    5    6    7    Out of control

Reason(s):

26. Under control    1    2    3    4    5    6    7    Out of control  
Reason(s):

27. Under control    1    2    3    4    5    6    7    Out of control  
Reason(s):

28. Under control    1    2    3    4    5    6    7    Out of control  
Reason(s):

29. Under control    1    2    3    4    5    6    7    Out of control  
Reason(s):

30. Under control    1   2   3   4   5   6   7   Out of control  
Reason(s):

31. Under control    1   2   3   4   5   6   7   Out of control  
Reason(s):

32. Under control    1   2   3   4   5   6   7   Out of control  
Reason(s):

33. Under control    1   2   3   4   5   6   7   Out of control  
Reason(s):

## 9 Playing from a control panel

The last task consists in controlling the pole and cart using a completely different interface. This is an interface that resembles a control panel, in which the angle, position and velocities that determine the state of the pole and cart are displayed using scales and sliders.

As before, you can control the pole and cart only by pushing the cart (or not) either to the left or to the right. You succeed as long as the pole angle is less than 90 degrees in any direction—that is, the pole has not reached a horizontal position—and the cart position is less than 2.4 metres in any direction—that is, the cart is still inside the window. Otherwise you have crashed the thing.

**Please do restart the task every time you crash.**

Input is restricted to pressing arrow keys: ↑ to restart the task, ← to push the cart to the left and → to push the cart to the right. A force of fixed magnitude will be applied to the cart in the appropriate direction for as long as you hold the key down.

You have five minutes for playing. **Please enjoy and do your best!**

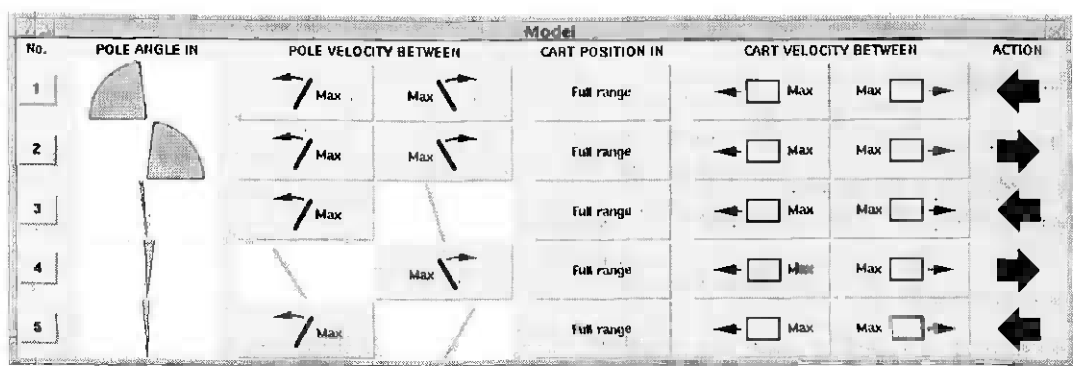


Appendix D

Learner models presented in  
Experiment 1

left	408	89	20	if $a \leq -0.135$ .
right	20	89	408	if $a \geq 0.135$ .
left	259	156	1	if $-0.135 \leq a \leq -0.075$ and $\dot{a} \leq 0.017$ .
right	9	388	627	if $-0.032 \leq a \leq 0.135$ and $\dot{a} \geq 0.210$ .
left	503	352	9	if $-0.075 \leq a \leq 0.036$ and $\dot{a} \leq -0.210$ .

(a) Learner model

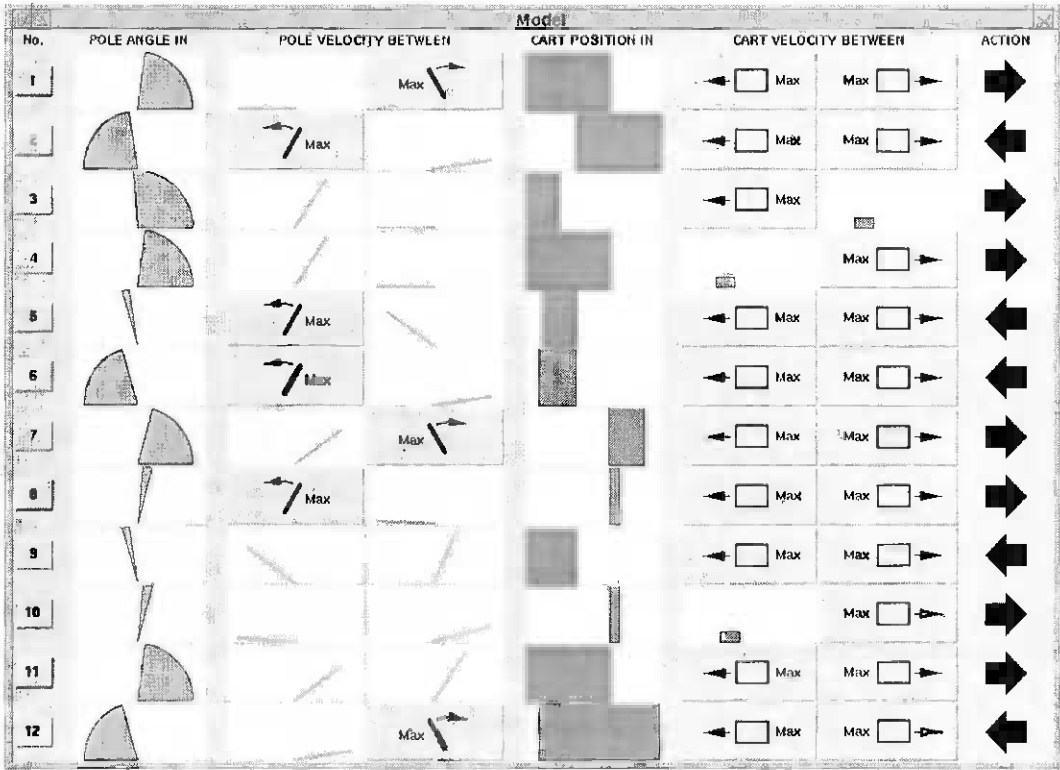


(b) Graphical presentation

Figure D.1: Learner model presented to Participant 1 in Experiment 1.

right	0	1	265	if $x \leq 0.564$ and $a \geq 0.142$ and $\dot{a} \geq 1.833$ .
left	816	8	2	if $x \geq -0.564$ and $a \leq -0.142$ and $\dot{a} \leq 0.311$ .
right	0	3	294	if $x \leq -1.214$ and $\dot{x} \leq -0.916$ and $a \geq -0.071$ and $-0.050 \leq \dot{a} \leq 1.833$ .
right	0	6	257	if $x \leq 0.564$ and $\dot{x} \geq -0.916$ and $a \geq 0.142$ and $-0.050 \leq \dot{a} \leq 1.833$ .
left	271	7	1	if $-1.732 \leq x \leq -0.611$ and $-0.275 \leq a \leq -0.153$ and $\dot{a} \leq 0.520$ .
left	204	6	0	if $-1.874 \leq x \leq -0.564$ and $a \leq -0.275$ and $\dot{a} \leq 0.311$ .
right	0	7	223	if $0.564 \leq x \leq 1.838$ and $a \geq 0.263$ and $\dot{a} \geq -0.520$ .
right	3	5	182	if $0.611 \leq x \leq 0.957$ and $0.092 \leq a \leq 0.263$ and $\dot{a} \leq 0.323$ .
left	65	1	2	if $x \leq -0.580$ and $-0.275 \leq a \leq -0.146$ and $0.520 \leq \dot{a} \leq 1.106$ .
right	1	7	128	if $0.611 \leq x \leq 0.957$ and $\dot{x} \geq -0.158$ and $0.092 \leq a \leq 0.263$ and $0.323 \leq \dot{a} \leq 1.600$ .
right	2	5	73	if $x \leq 0.564$ and $a \geq 0.142$ and $-0.520 \leq \dot{a} \leq -0.050$ .
left	94	13	3	if $x \geq -1.874$ and $a \leq -0.275$ and $\dot{a} \geq 0.311$ .

(a) Learner model



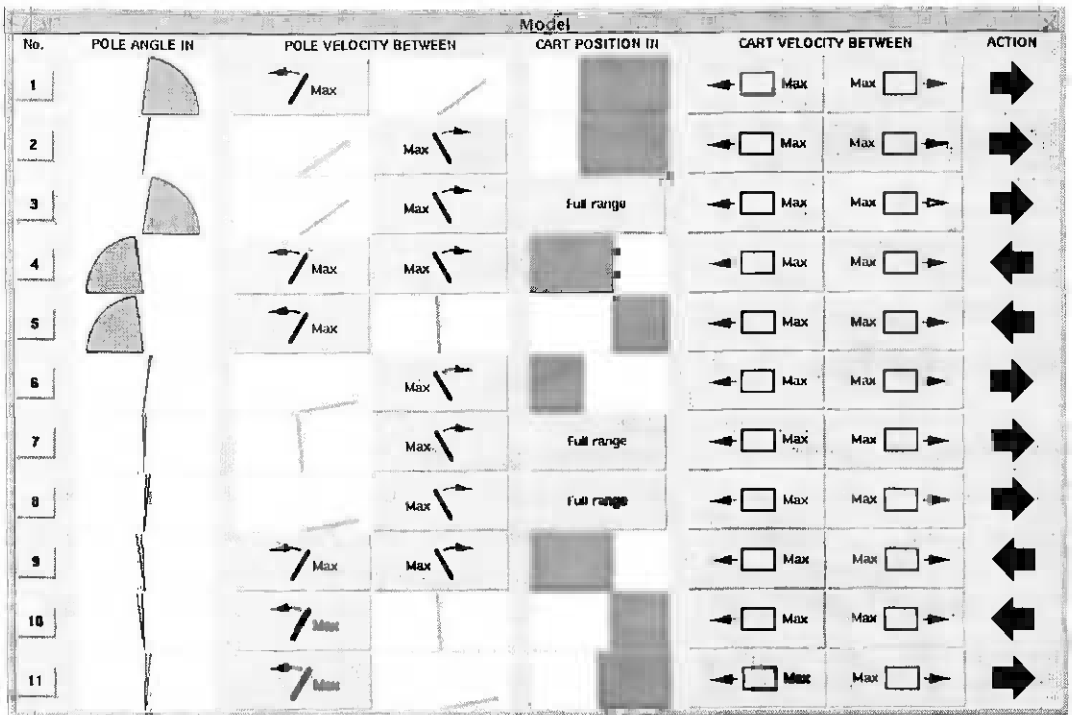
(b) Graphical presentation

Figure D.2: Learner model presented to Participant 4 in Experiment 1.



right	3	1	77	if $x \geq -0.561$ and $a \geq 0.122$ and $\dot{a} \leq -0.029$ .
right	1	3	61	if $x \geq -0.561$ and $0.122 \leq a \leq 0.163$ and $\dot{a} \geq -0.029$ .
right	3	52	380	if $a \geq 0.163$ and $\dot{a} \geq -0.029$ .
left	183	22	6	if $x \leq 0.555$ and $a \leq -0.126$ .
left	375	71	1	if $x \geq 0.555$ and $a \leq -0.126$ and $\dot{a} \leq 0.075$ .
right	1	11	40	if $x \leq -0.561$ and $0.122 \leq a \leq 0.163$ and $\dot{a} \geq 0.146$ .
right	4	25	69	if $-0.019 \leq a \leq 0.047$ and $\dot{a} \geq 0.228$ .
right	10	34	81	if $0.047 \leq a \leq 0.122$ and $\dot{a} \geq 0.146$ .
left	99	40	30	if $x \leq 0.456$ and $-0.126 \leq a \leq -0.039$ .
left	78	55	9	if $x \geq 0.456$ and $-0.126 \leq a \leq -0.039$ and $\dot{a} \leq 0.075$ .
right	29	15	49	if $x \geq -0.043$ and $0.025 \leq a \leq 0.122$ and $\dot{a} \leq 0.146$ .

(a) Learner model

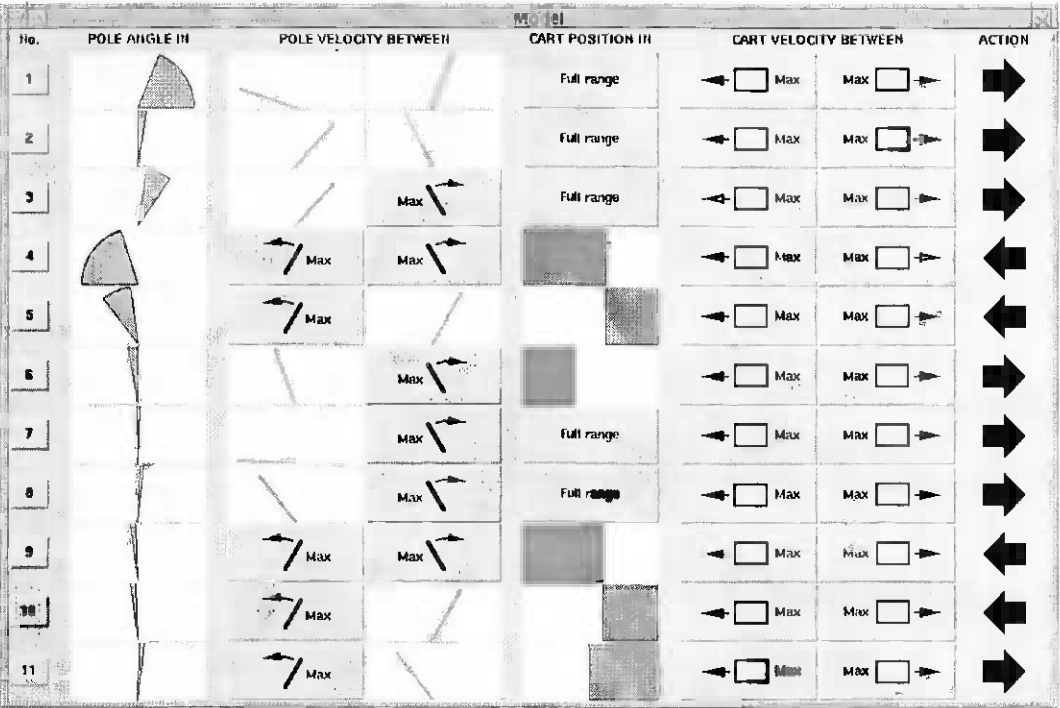


(b) Graphical presentation

Figure D.3: Learner model presented to Participant 6 in Experiment 1.

right	0	9	131	if $a \geq 0.383$ and $0.125 \leq \dot{a} \leq 1.375$ .	
right	7	124	171	if $0.004 \leq a \leq 0.163$ and $-0.029 \leq \dot{a} \leq 0.500$ .	
right	2	28	312	if $0.163 \leq a \leq 0.599$ and $\dot{a} \geq -0.029$ .	
left	24	17	1	if $x \leq 0.555$ and $a \leq -0.310$ .	
left	325	62	1	if $x \geq 0.555$ and $-0.666 \leq a \leq -0.126$ and $\dot{a} \leq 0.075$ .	
right	148	29	44	if $x \leq -0.561$ and $-0.171 \leq a \leq 0.004$ and $\dot{a} \geq -1.375$ .	
right	4	25	69	if $-0.019 \leq a \leq 0.047$ and $\dot{a} \geq 0.228$ .	
right	10	34	81	if $0.047 \leq a \leq 0.122$ and $\dot{a} \geq 0.146$ .	
left	99	40	30	if $x \leq 0.456$ and $-0.126 \leq a \leq -0.039$ .	
left	78	55	9	if $x \geq 0.456$ and $-0.126 \leq a \leq -0.039$ and $\dot{a} \leq 0.075$ .	
right	29	15	49	if $x \geq -0.043$ and $0.025 \leq a \leq 0.122$ and $\dot{a} \leq 0.146$ .	

(a) Learner model

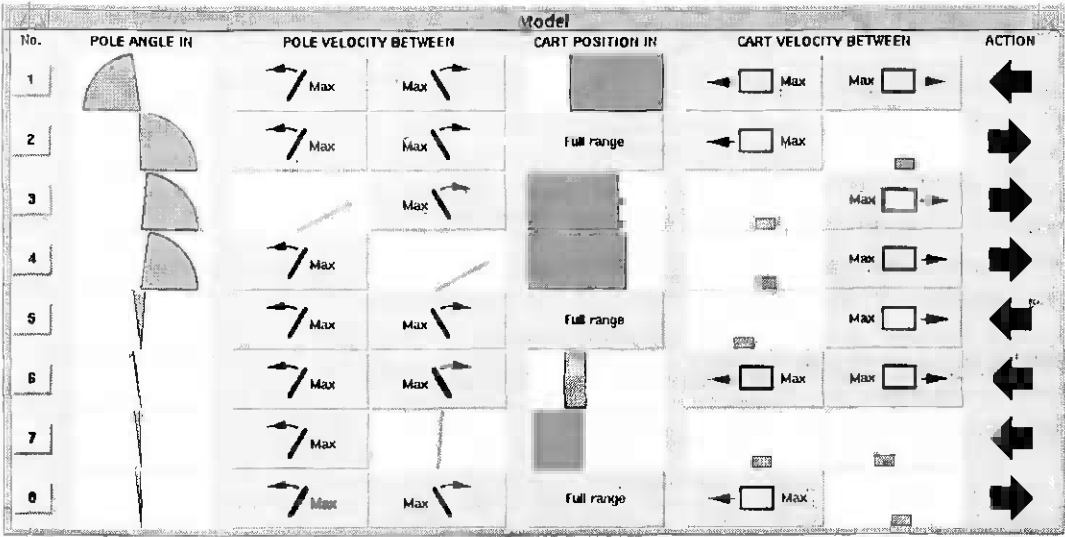


(b) Graphical presentation

Figure D.4: The learner model of Participant 6 after being edited.

left 554 72 15 if  $x \geq -0.861$  and  $a \leq -0.147$  .  
 right 3 34 228 if  $\dot{x} \leq -1.280$  and  $a \geq -0.024$  .  
 right 15 71 398 if  $x \leq 0.817$  and  $\dot{x} \geq -1.280$  and  $a \geq 0.110$  and  $\dot{a} \geq -0.014$  .  
 right 14 41 151 if  $x \leq 1.090$  and  $\dot{x} \geq -1.280$  and  $a \geq 0.139$  and  $\dot{a} \leq -0.014$  .  
 left 148 51 4 if  $\dot{x} \geq 1.425$  and  $-0.120 \leq a \leq 0.078$  .  
 left 50 18 4 if  $-1.075 \leq x \leq -0.330$  and  $-0.147 \leq a \leq -0.120$  .  
 left 43 21 5 if  $-2.163 \leq x \leq -0.506$  and  $-1.821 \leq \dot{x} \leq 1.425$   
 and  $-0.120 \leq a \leq -0.034$  and  $\dot{a} \leq 0.150$  .  
 right 5 23 43 if  $\dot{x} \leq -1.841$  and  $-0.112 \leq a \leq -0.024$  .

(a) Learner model

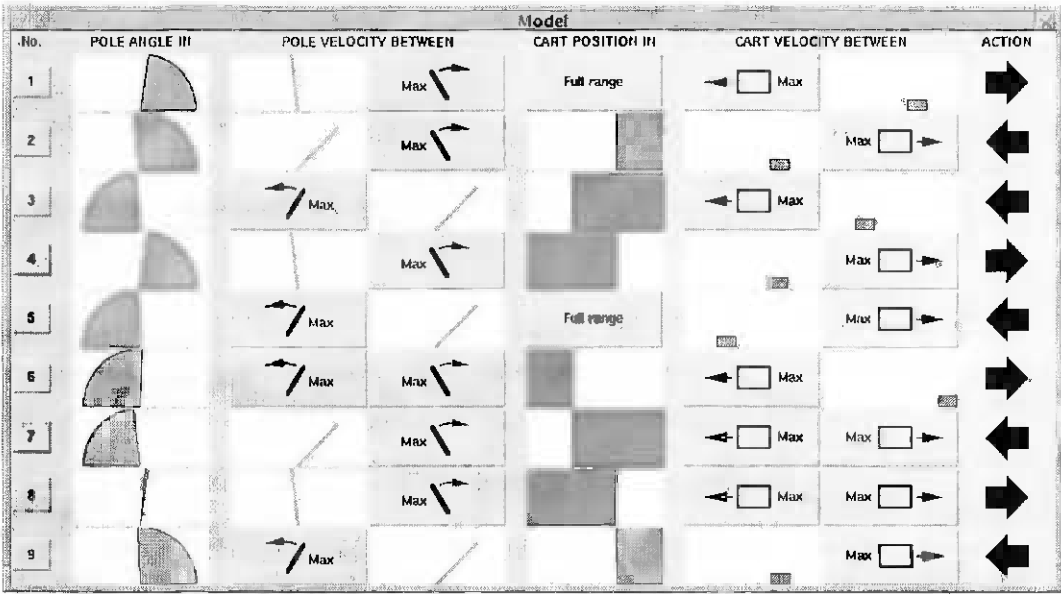


(b) Graphical presentation

Figure D.5: Learner model presented to Participant 10 in Experiment 1.

right	7	64	477	if $\dot{x} \leq 1.663$ and $a \geq 0.148$ and $\dot{a} \geq -0.195$ .
left	61	9	1	if $x \geq 0.785$ and $\dot{x} \geq 2.398$ and $a \geq -0.100$ and $\dot{a} \geq -0.174$ .
left	92	16	1	if $x \geq -0.762$ and $\dot{x} \leq -1.663$ and $a \leq -0.100$ and $\dot{a} \leq -0.174$ .
right	2	22	126	if $x \leq 0.749$ and $\dot{x} \geq 1.663$ and $a \geq 0.148$ and $\dot{a} \geq -0.195$ .
left	500	106	12	if $\dot{x} \geq -1.663$ and $a \leq -0.038$ and $\dot{a} \leq -0.174$ .
right	12	42	107	if $x \leq -0.780$ and $\dot{x} \leq -2.049$ and $a \leq 0.043$ .
left	223	116	20	if $x \geq -0.762$ and $a \leq -0.100$ and $\dot{a} \geq -0.174$ .
right	6	64	98	if $x \leq 0.749$ and $0.100 \leq a \leq 0.148$ and $\dot{a} \geq -0.195$ .
left	33	23	3	if $x \geq 0.785$ and $\dot{x} \geq 2.398$ and $a \geq -0.038$ and $\dot{a} \leq -0.174$ .

(a) Learner model

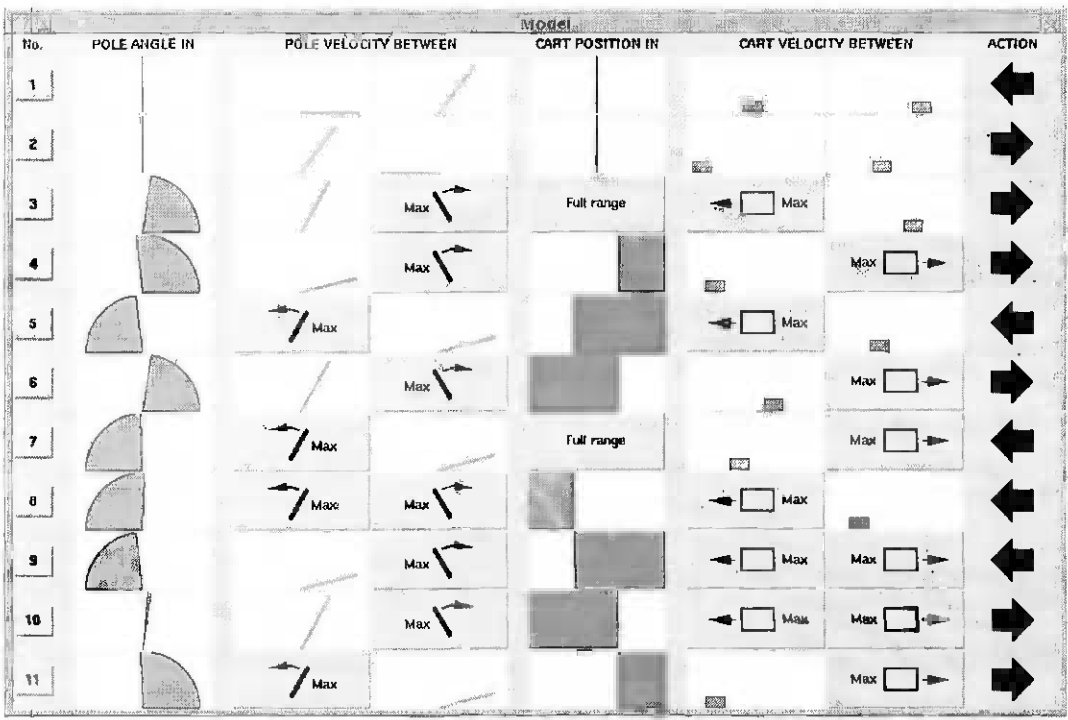


(b) Graphical presentation

Figure D.6: Learner model presented to Participant 15 in Experiment 1.

left	0	0	0	if $0.000 \leq x \leq 0.000$ and $-0.844 \leq \dot{x} \leq -0.422$ and $0.000 \leq a \leq 0.000$ and $-0.250 \leq \dot{a} \leq -0.025$ .
right	0	0	0	if $0.000 \leq x \leq 0.000$ and $0.141 \leq \dot{x} \leq 0.562$ and $0.000 \leq a \leq 0.000$ and $-0.025 \leq \dot{a} \leq 0.250$ .
right	7	64	477	if $\dot{x} \leq 1.663$ and $a \geq 0.148$ and $\dot{a} \geq -0.195$ .
right	61	9	1	if $x \geq 0.785$ and $\dot{x} \geq 2.398$ and $a \geq -0.100$ and $\dot{a} \geq -0.174$ .
left	92	16	1	if $x \geq -0.762$ and $\dot{x} \leq -1.663$ and $a \leq -0.100$ and $\dot{a} \leq -0.174$ .
right	2	22	126	if $x \leq 0.749$ and $\dot{x} \geq 1.663$ and $a \geq 0.148$ and $\dot{a} \geq -0.195$ .
left	500	106	12	if $\dot{x} \geq -1.663$ and $a \leq -0.038$ and $\dot{a} \leq -0.174$ .
left	12	42	107	if $x \leq -0.780$ and $\dot{x} \leq -2.049$ and $a \leq 0.043$ .
left	223	116	20	if $x \geq -0.762$ and $a \leq -0.100$ and $\dot{a} \geq -0.174$ .
right	6	64	98	if $x \leq 0.749$ and $0.100 \leq a \leq 0.148$ and $\dot{a} \geq -0.195$ .
right	33	23	3	if $x \geq 0.785$ and $\dot{x} \geq 2.398$ and $a \geq -0.038$ and $\dot{a} \leq -0.174$ .

(a) Learner model

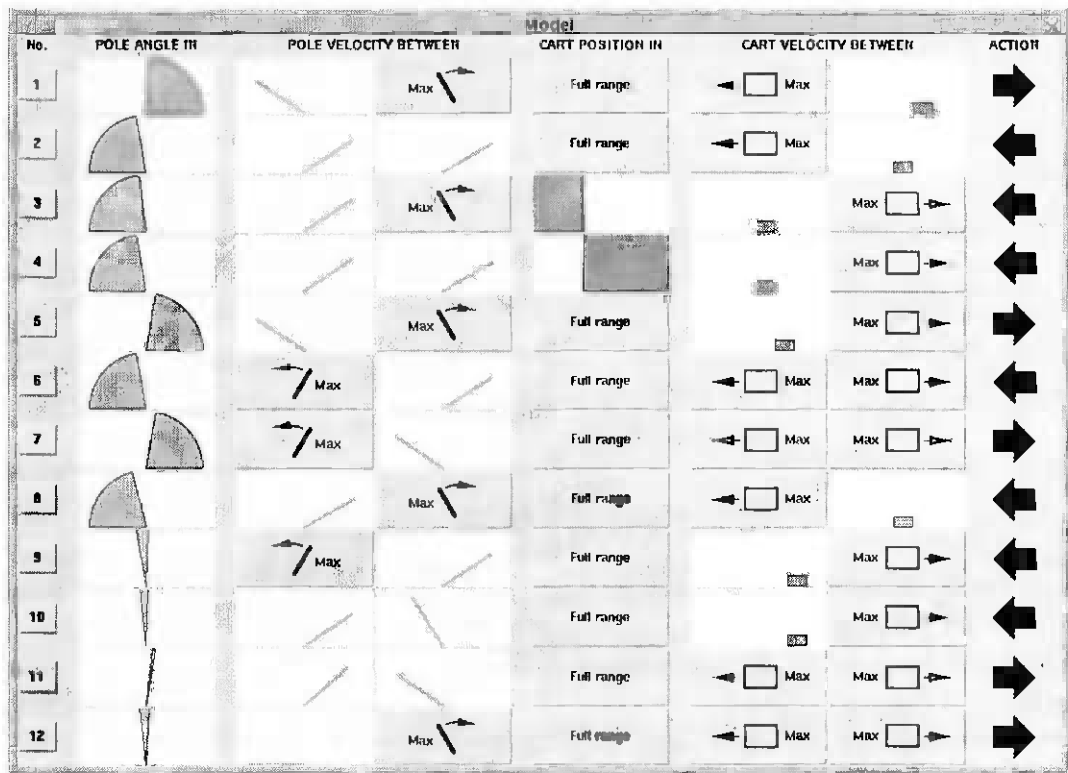


(b) Graphical presentation

Figure D.7: The learner model of Participant 15 after being edited.

right	9	13	331	if $\dot{x} \leq 0.143$ and $a \geq 0.066$ and $\dot{a} \geq 0.164$ .
left	72	4	0	if $\dot{x} \leq -2.232$ and $a \leq -0.159$ and $-0.982 \leq \dot{a} \leq 0.527$ .
left	389	27	9	if $\dot{x} \leq -0.595$ and $\dot{x} \geq -2.232$ and $a \leq -0.110$ and $\dot{a} \geq -0.982$ .
left	217	21	1	if $\dot{x} \geq -0.595$ and $\dot{x} \geq -2.232$ and $a \leq -0.159$ and $-0.982 \leq \dot{a} \leq 0.527$ .
right	17	43	394	if $\dot{x} \geq 0.143$ and $a \geq 0.174$ and $\dot{a} \geq 0.164$ .
left	290	33	20	if $a \leq -0.159$ and $\dot{a} \leq -0.982$ .
right	19	37	261	if $a \geq 0.174$ and $\dot{a} \leq 0.164$ .
left	38	6	2	if $\dot{x} \leq -2.232$ and $a \leq -0.237$ and $\dot{a} \geq 0.527$ .
left	79	24	0	if $\dot{x} \geq 0.373$ and $-0.159 \leq a \leq 0.032$ and $\dot{a} \leq -0.982$ .
left	49	15	6	if $\dot{x} \geq 0.373$ and $-0.110 \leq a \leq 0.032$ and $-0.982 \leq \dot{a} \leq -0.093$ .
right	6	39	75	if $0.110 \leq a \leq 0.174$ and $-0.726 \leq \dot{a} \leq 0.164$ .
right	10	56	105	if $-0.110 \leq a \leq 0.066$ and $\dot{a} \geq 0.692$ .

(a) Learner model.

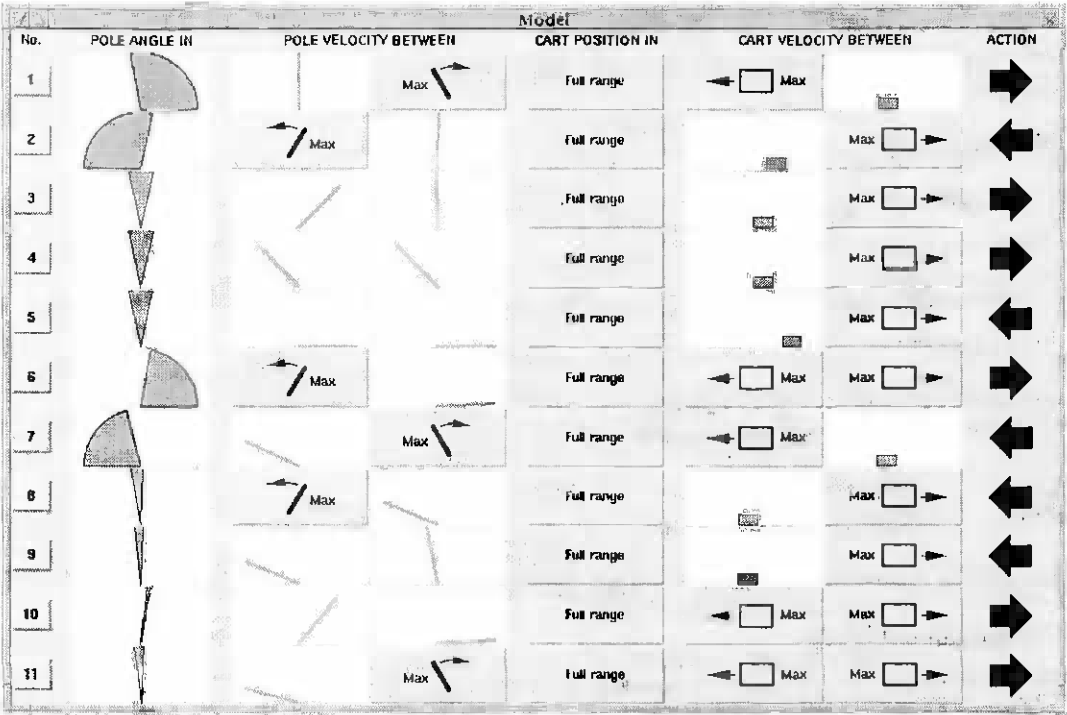


(b) Graphical presentation

Figure D.8: Learner model presented to Participant 19 in Experiment 1.

right	65	85	415	if $\dot{x} \leq -0.141$ and $a \geq -0.186$ and $\dot{a} \geq 0.250$ .
left	394	79	74	if $\dot{x} \geq 0.422$ and $a \leq 0.205$ and $\dot{a} \leq -0.250$ .
right	13	16	31	if $\dot{x} \geq 0.143$ and $-0.208 \leq a \leq 0.235$ and $0.125 \leq \dot{a} \leq 0.250$ .
right	0	0	0	if $\dot{x} \geq 0.143$ and $-0.208 \leq a \leq 0.235$ and $-0.125 \leq \dot{a} \leq -0.125$ .
left	0	0	0	if $\dot{x} \geq -1.406$ and $-0.197 \leq a \leq 0.209$ and $0.000 \leq \dot{a} \leq 0.000$ .
right	19	37	261	if $a \geq 0.174$ and $\dot{a} \leq 0.164$ .
left	38	6	2	if $\dot{x} \leq -2.232$ and $a \leq -0.237$ and $\dot{a} \geq 0.527$ .
left	79	24	0	if $\dot{x} \geq 0.373$ and $-0.159 \leq a \leq 0.032$ and $\dot{a} \leq -0.982$ .
left	49	15	6	if $\dot{x} \geq 0.373$ and $-0.110 \leq a \leq 0.032$ and $-0.982 \leq \dot{a} \leq -0.093$ .
right	6	39	75	if $0.110 \leq a \leq 0.174$ and $-0.726 \leq \dot{a} \leq 0.164$ .
right	10	56	105	if $-0.110 \leq a \leq 0.066$ and $\dot{a} \geq 0.692$ .

(q) Learner model

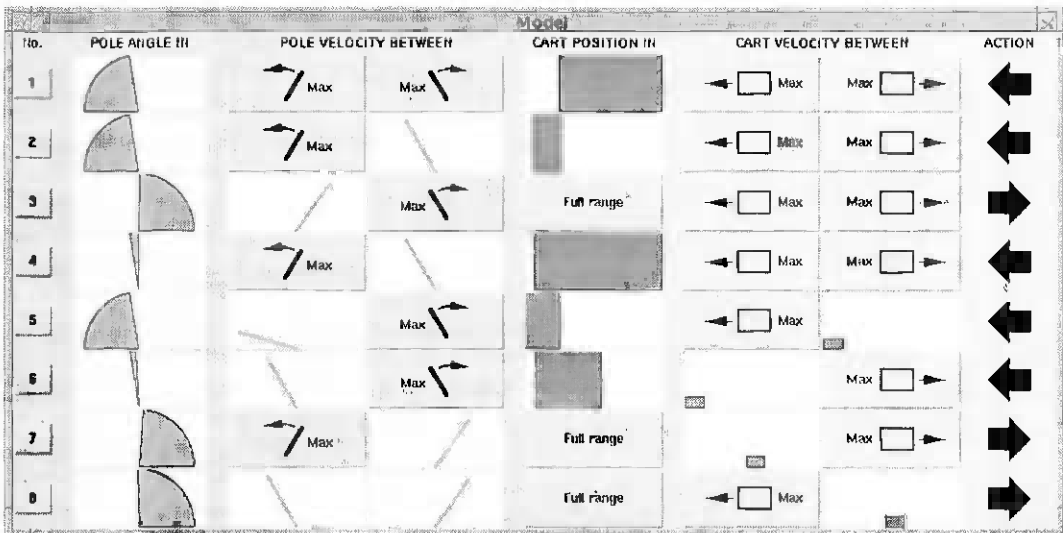


(b) Graphical presentation

Figure D.9: The learner model of Participant 19 after being edited.

left	595	50	4	if $x \geq -1.183$ and $a \leq -0.163$ .
left	53	0	0	if $-2.075 \leq x \leq -1.183$ and $a \leq -0.163$ and $\dot{a} \leq -0.232$ .
right	9	101	670	if $a \geq 0.022$ and $\dot{a} \geq 0.235$ .
left	233	53	11	if $x \geq -2.075$ and $-0.163 \leq a \leq 0.009$ and $\dot{a} \leq -0.232$ .
left	32	5	3	if $x \leq -1.183$ and $\dot{x} \leq -1.854$ and $a \leq -0.163$ and $\dot{a} \geq -0.166$ .
left	98	28	3	if $-2.021 \leq x \leq 0.269$ and $\dot{x} \geq -1.854$ and $-0.163 \leq a \leq -0.044$ and $\dot{a} \geq -0.232$ .
right	25	93	318	if $\dot{x} \geq -0.728$ and $a \geq 0.083$ and $\dot{a} \leq 0.235$ .
right	4	18	58	if $\dot{x} \leq -0.728$ and $a \geq -0.025$ and $-0.232 \leq \dot{a} \leq 0.235$ .

(a) Learner model



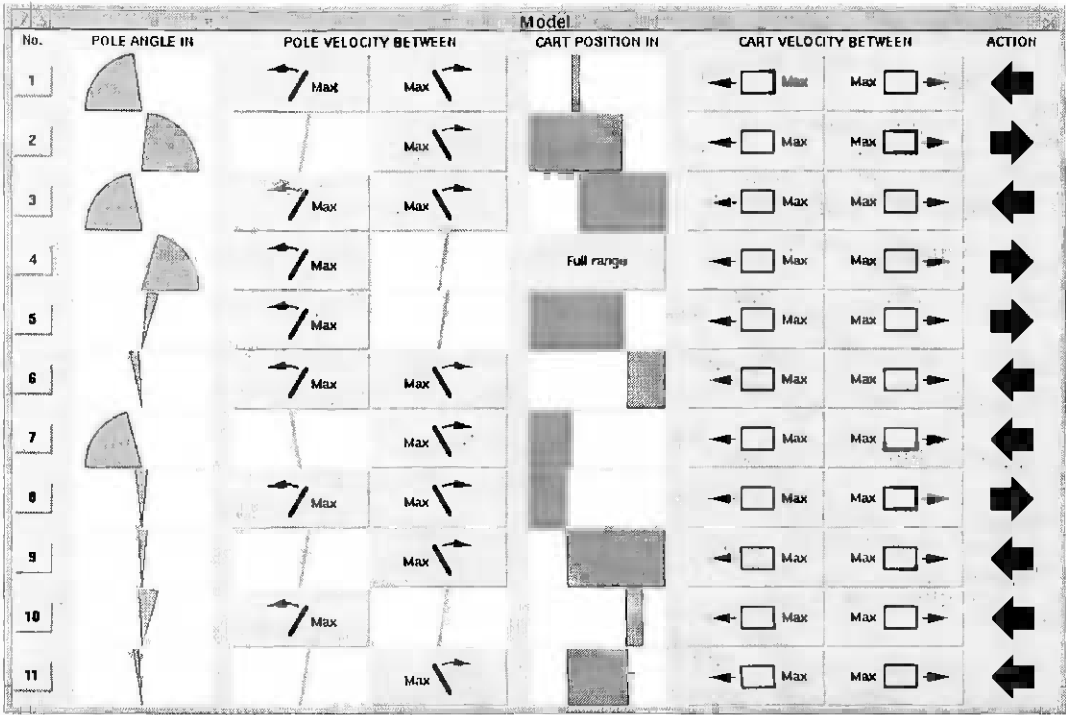
(b) Graphical presentation

Figure D.10: Learner model presented to Participant 20 in Experiment 1



left 393 12 0 if  $-0.857 \leq x \leq -0.606$  and  $a \leq -0.153$  .  
right 0 8 158 if  $x \leq 0.911$  and  $a \geq 0.097$  and  $\dot{a} \geq 2.161$  .  
left 508 34 9 if  $x \geq -0.606$  and  $a \leq -0.201$  .  
right 31 39 450 if  $a \geq 0.290$  and  $\dot{a} \leq 2.161$  .  
right 17 114 553 if  $x \leq 0.911$  and  $0.097 \leq a \leq 0.290$  and  $\dot{a} \leq 2.161$  .  
left 134 37 2 if  $x \geq 1.092$  and  $-0.201 \leq a \leq -0.047$  .  
left 84 20 30 if  $x \leq -0.937$  and  $a \leq -0.253$  and  $\dot{a} \geq -2.161$  .  
right 12 63 124 if  $x \leq -1.094$  and  $-0.118 \leq a \leq 0.097$  .  
left 46 24 12 if  $x \geq -0.990$  and  $-0.047 \leq a \leq 0.097$  and  $\dot{a} \geq -0.441$  .  
left 44 36 3 if  $1.078 \leq x \leq 1.641$  and  $-0.047 \leq a \leq 0.290$  and  $\dot{a} \leq -0.441$  .  
left 53 38 12 if  $-0.990 \leq x \leq 1.092$   
and  $-0.153 \leq a \leq -0.047$  and  $\dot{a} \geq -0.441$  .

(a) Learner model

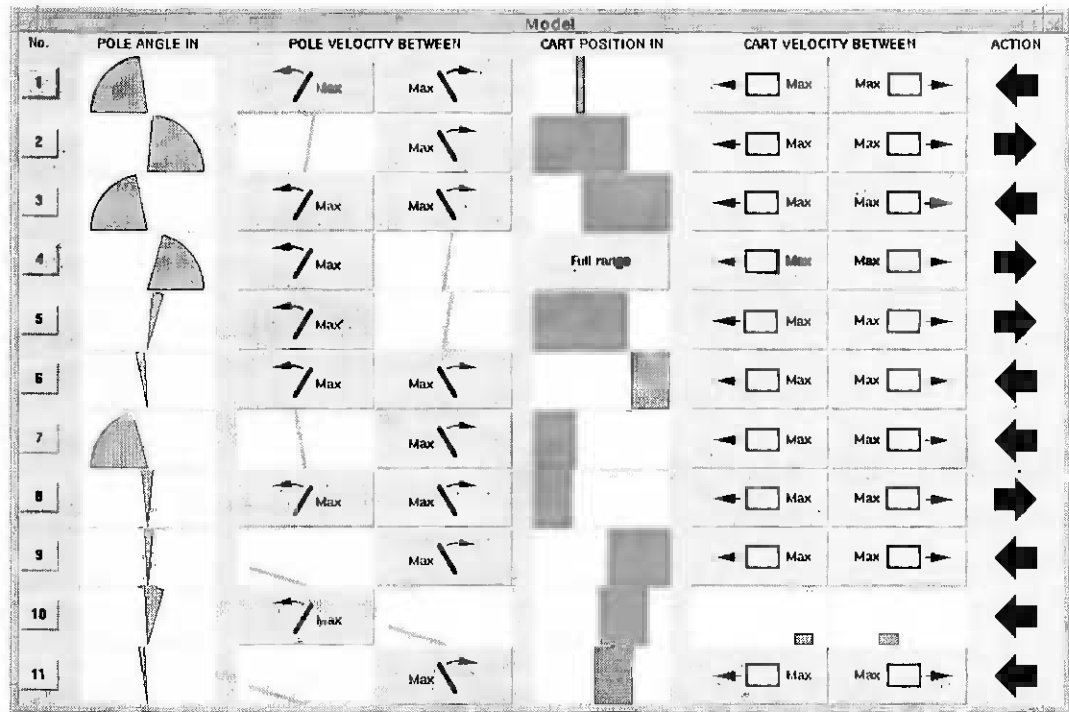


(b) Graphical presentation

Figure D.11: Learner model presented to Participant 23 in Experiment 1.

left	393	12	0	if $-0.857 \leq x \leq -0.606$ and $a \leq -0.153$ .
right	0	8	158	if $x \leq 0.911$ and $a \geq 0.097$ and $\dot{a} \geq 2.161$ .
left	508	34	9	if $x \geq -0.606$ and $a \leq -0.201$ .
right	31	39	450	if $a \geq 0.290$ and $\dot{a} \leq 2.161$ .
right	17	114	553	if $x \leq 0.911$ and $0.097 \leq a \leq 0.290$ and $\dot{a} \leq 2.161$ .
left	134	37	2	if $x \geq 1.092$ and $-0.201 \leq a \leq -0.047$ .
left	84	20	30	if $x \leq -0.937$ and $a \leq -0.253$ and $\dot{a} \geq -2.161$ .
right	12	63	124	if $x \leq -1.094$ and $-0.118 \leq a \leq 0.097$ .
left	46	24	12	if $x \geq 0.300$ and $-0.047 \leq a \leq 0.097$ and $\dot{a} \geq -0.441$ .
left	0	0	0	if $0.000 \leq x \leq 1.628$ and $-2.250 \leq \dot{x} \leq -1.266$ and $-0.047 \leq a \leq 0.290$ and $\dot{a} \leq -0.441$ .
left	0	0	0	if $-0.225 \leq x \leq 1.092$ and $-0.153 \leq a \leq -0.047$ and $\dot{a} \geq -0.441$ .

(a) Learner model

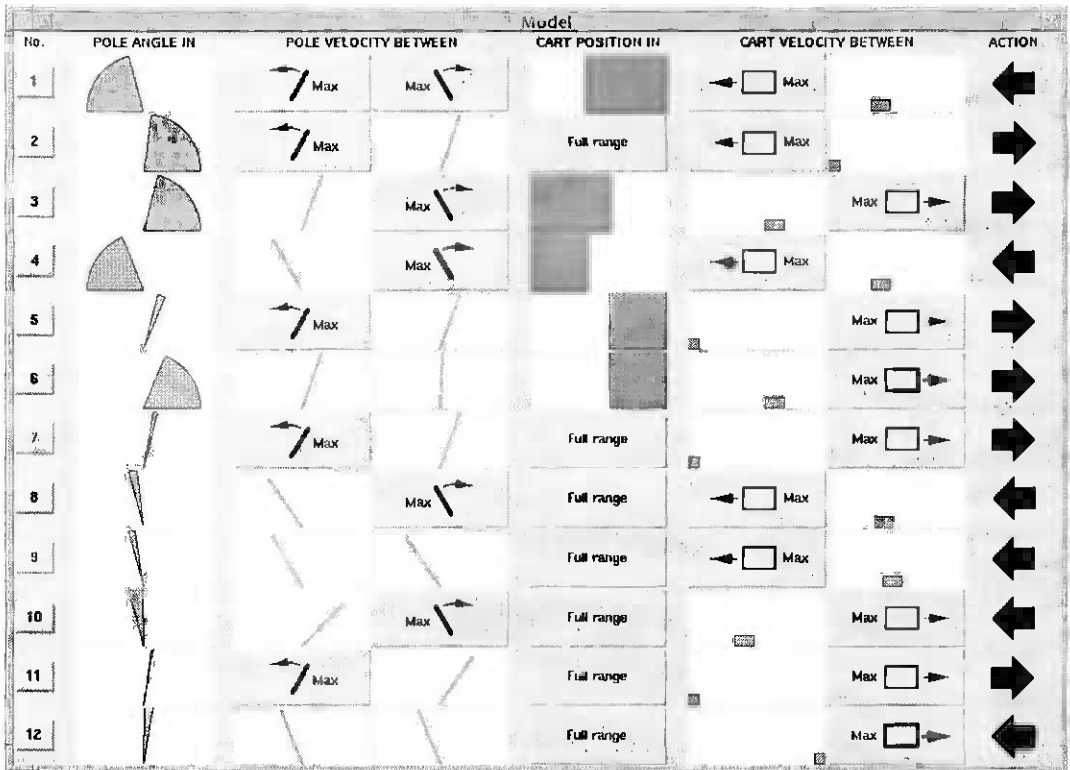


(b) Graphical presentation

Figure D.12: The learner model of Participant 23 after being edited.

left	94	3	5	if $x \geq -0.425$ and $\dot{x} \leq 2.261$ and $a \leq -0.262$ .
right	2	7	91	if $\dot{x} \leq 0.897$ and $a \geq 0.143$ and $\dot{a} \leq 0.507$ .
right	5	0	51	if $x \leq 0.425$ and $\dot{x} \geq -2.259$ and $a \geq 0.246$ and $\dot{a} \geq 0.507$ .
left	80	8	6	if $x \leq -0.425$ and $\dot{x} \leq 2.261$ and $a \leq -0.390$ and $\dot{a} \geq -2.616$ .
right	1	5	38	if $x \geq 0.425$ and $\dot{x} \geq 0.897$ and $0.246 \leq a \leq 0.392$ and $\dot{a} \leq 0.507$ .
right	7	9	85	if $x \geq 0.425$ and $\dot{x} \geq -2.259$ and $a \geq 0.392$ and $0.507 \leq \dot{a} \leq 2.698$ .
right	1	9	56	if $\dot{x} \geq 0.897$ and $0.178 \leq a \leq 0.246$ and $\dot{a} \leq 0.507$ .
left	167	31	18	if $\dot{x} \leq -0.477$ and $-0.262 \leq a \leq -0.110$ and $\dot{a} \geq -0.330$ .
left	55	14	5	if $\dot{x} \leq -0.753$ and $-0.262 \leq a \leq -0.146$ and $-2.616 \leq \dot{a} \leq -0.330$ .
left	57	6	14	if $\dot{x} \geq -0.477$ and $-0.262 \leq a \leq -0.020$ and $\dot{a} \geq -1.309$ .
right	8	13	58	if $\dot{x} \geq 0.897$ and $0.143 \leq a \leq 0.178$ and $\dot{a} \leq 0.330$ .
left	36	3	10	if $\dot{x} \geq 0.291$ and $0.029 \leq a \leq 0.178$ and $0.667 \leq \dot{a} \leq 1.476$ .

(a) Learner model

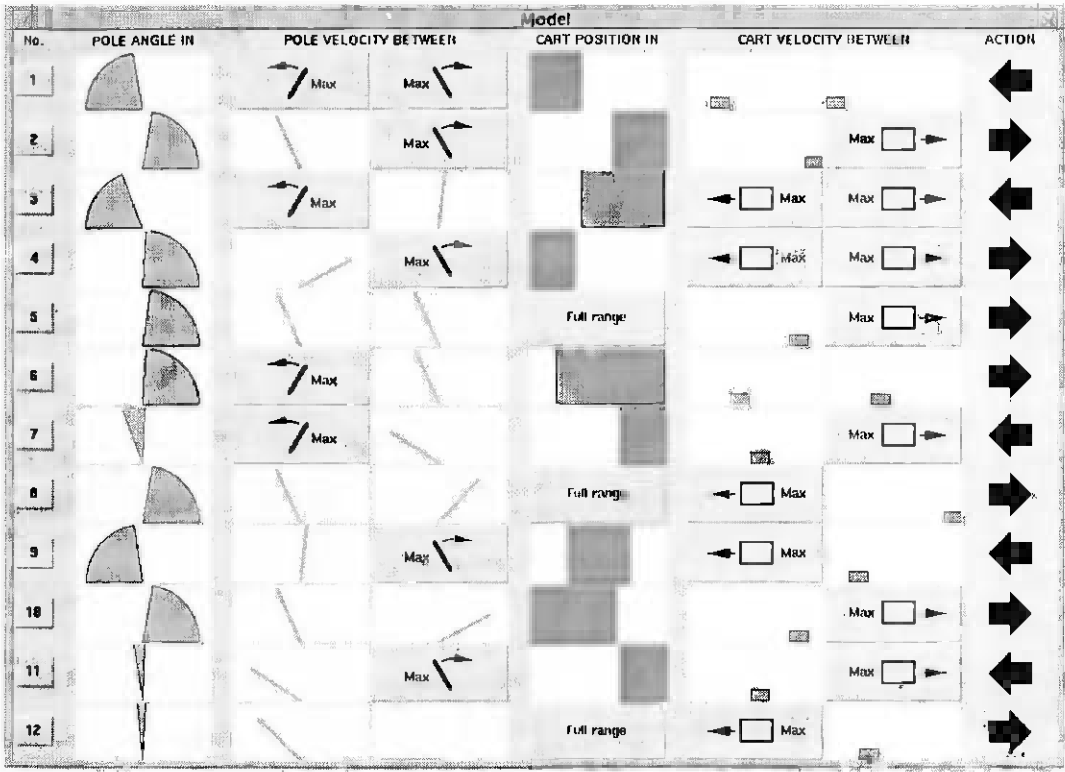


(b) Graphical presentation

Figure D.13: Learner model presented to Participant 27 in Experiment 1.

left	345	4	0	if $x \leq -0.623$ and $-2.034 \leq \dot{x} \leq -0.573$ and $a \leq -0.169$ .
right	2	4	287	if $x \geq 0.623$ and $\dot{x} \geq 0.573$ and $a \geq 0.169$ and $\dot{a} \geq -0.123$ .
left	368	6	5	if $x \geq -0.517$ and $a \leq -0.362$ and $\dot{a} \leq 0.104$ .
right	6	10	464	if $x \leq -0.861$ and $a \geq 0.057$ and $\dot{a} \geq 0.358$ .
right	2	3	61	if $\dot{x} \geq 0.850$ and $a \geq 0.144$ and $-0.320 \leq \dot{a} \leq -0.123$ .
right	4	5	87	if $x \geq -1.404$ and $0.714 \leq \dot{x} \leq 2.395$ and $a \geq 0.058$ and $\dot{a} \leq -0.320$ .
left	209	21	5	if $x \geq 0.853$ and $\dot{x} \geq 1.060$ and $-0.362 \leq a \leq 0.037$ and $\dot{a} \leq -0.356$ .
right	1	4	48	if $\dot{x} \leq 0.573$ and $a \geq 0.169$ and $-0.123 \leq \dot{a} \leq 0.339$ .
left	88	5	6	if $-0.990 \leq x \leq 1.078$ and $\dot{x} \leq -2.034$ and $a \leq -0.148$ and $\dot{a} \geq 0.104$ .
right	0	7	56	if $x \leq 0.623$ and $\dot{x} \geq 0.850$ and $a \geq 0.169$ and $-0.123 \leq \dot{a} \leq 0.358$ .
left	42	8	2	if $x \geq 0.853$ and $\dot{x} \geq 1.060$ and $-0.148 \leq a \leq 0.037$ and $\dot{a} \geq -0.356$ .
right	0	11	55	if $\dot{x} \leq -1.092$ and $-0.090 \leq a \leq 0.057$ and $0.652 \leq \dot{a} \leq 1.830$ .

(a) Learner model



(b) Graphical presentation

Figure D.14: Learner model presented to Participant 29 in Experiment 1.

## D.1 Example of inspecting and editing of a learner model

This transcription of the **Condition** stage of Experiment 1 for Participant 6 is included here as an example of the inspecting and editing activity the participants in the experiment engaged in.

### D.1.1 Inspecting the learner model

Verbalise rule 3

Justify rule 5

Justify rule 9

Justify rule 11

Run rule 8 (twice)

Run rule 9

Run rule 10

Run rule 11

Run rule 6

With the exceptions of Rules 3 and 5, the learner seems to concentrate on inspecting the last six rules in the model, which have tight constraints on pole angle. Rules 3 and 5 are very similar to Rules 1 and 4, respectively, so it is possible the inclusion of the former in the model intrigued the learner.

### D.1.2 Editing the learner model

Change position in rule 1 (three times)

The learner removes the precondition on cart position from Rule 1.

Change position in rule 2 (twice)

The precondition on cart position in Rule 2 is also removed.

Change angle in rule 1

The range of pole angles that match Rule 1 is narrowed by increasing the lower limit.

Change angle in rule 2

The learner makes the precondition on pole angle to start from the vertical (what can be seen as a more natural precondition than the one in the original model).

Change angle in rule 3

By now the preconditions on pole angle of Rules 1 to 3 are roughly supplementary, covering all positive angles.

Raise `angvelmin` in rule 1

Change `angvelmin` in rule 1

Change `angvelmax` in rule 1

Change `angvelmin` in rule 1

There is no lower limit in the precondition on pole velocity of Rule 1, so the icon representing maximum anti-clockwise speed is displayed instead. The learner first uncovers the animation behind the icon (featuring a high anti-clockwise speed) and then makes its velocity equal to the upper limit of the same precondition (discovering in the way that PACMOD does not allow him to set a lower limit higher than the upper limit). He adjusts the upper limit, and then comes back to a final adjustment of the lower limit.

Raise `angvelmax` in rule 2

Change `angvelmax` in rule 2

The range of pole velocities that matches Rule 2 is narrowed.

Run rule 1 (three times)

Run rule 2 (seven times)

Run rule 1 (nine times)

The learner verifies the changes made to Rules 1 and 2. He does not verify the modifications made to Rule 3.

Change `angle` in rule 5

Change `angle` in rule 4

Change `angle` in rule 5

Change `angle` in rule 6 (twice)

By now the preconditions on pole angle of Rules 4 to 6 are roughly symmetrical to the preconditions of Rules 1 to 3.

Change `angvelmin` in rule 6 (twice)

Time over

The learner starts modifying the preconditions on pole velocity of Rules 4 to 6, probably with the intention of making them symmetrical to the corresponding preconditions of Rules 1 to 3. Unfortunately, he runs out of time ...

## Appendix E

### Published papers

The following papers have been published which are directly related to the research described in this dissertation. They are reproduced here with permission.

Morales, R., Ramscar, M. & Pain, H. (1998). Cognitive Effects of Participative Learner Modelling. In *Proceedings of the Current Trends and Applications of Artificial Intelligence in Education Workshop*, G. Ayala, ed., ITESM, Mexico City, Mexico. pp. 49–56.

Morales, R. & Pain, H. (1999). Modelling of novices' control skills with machine learning. In *UM99 User Modeling: Proceedings of the Seventh International Conference*, J. Kay, ed., Springer Wien New York. pp. 159–168.

Morales, R., Pain, H. & Conlon, T. (2000). Understandable learner models for a sensorimotor control task. In *Intelligent Tutoring Systems: Fifth International Conference, ITS'2000*, G. Gauthier, C. Frasson & K. VanLehn, eds., no. 1839 in Lecture Notes in Computer Science. Springer-Verlag, Montréal, Canada. pp. 222–231.





## Cognitive Effects of Participative Learner Modelling

Rafael Morales

R.Morales@ed.ac.uk

Michael Ramscar

M.Ramscar@ed.ac.uk

Helen Pain

H.Pain@ed.ac.uk

Department of Artificial Intelligence

University of Edinburgh

80 South Bridge, Edinburgh EH1 1HN, Scotland, UK

Tel: +44 (131) 650 2725

Fax: +44 (131) 650 6516

**Abstract.** The effect of *participative learner modelling*—a learner modelling process characterised by active and explicit participation of the learner—on the cognitive state of the learner and other participants is discussed. Two consequences of using participative learner modelling suggested are: (1) the acquisition by the learner of declarative representations of knowledge, as opposed to procedural encodings of it; and (2) the formation of common knowledge among the participants in the learner modelling task.

### 1 Introduction

One of the most widespread applications of user modelling has been in the construction of intelligent systems for supporting teaching and learning, in which the user is considered to be a *learner* of some kind of knowledge or skill, and the *learner model* is usually a dynamic representation of the emerging skill and knowledge of the user.

Traditionally such models have been considered part of the system internal information; they have been kept hidden from and inaccessible to the user. By contrast, *participative learner modelling* (PLM) is characterised by opening the learner model to the learner's inspection and direct influence on its content. This change of attitude has been motivated by two objectives:

1. To cope with the complexity of learner modelling, and hence to achieve more faithful learner models (Self, 1988).
2. To use learner models as educational tools, by *promoting the learner's reflection on, and awareness of her own knowledge* (Bull et al., 1995; Bull and Pain, 1995; Paiva et al., 1995; Self, 1988, 1994)

The latter aim presupposes that the involvement of the user in learner modelling is an educational task in itself, which does not distract the user from her primary objective of learning because she actually learns when dealing with the model (Bull, 1997; Cumming and Self, 1991). Some empirical evidence has been reported in support of that assumption (for example, in Cook and Kay, 1994).

The cognitive context encouraged by systems supporting PLM is different from the cognitive context promoted by systems with a covert learner model in at least two aspects: firstly, the learner is aware not only of the existence of a learner model—explicit, individualised, and dynamic—but also of the content of the model; and secondly, the learner's activities of inspecting and modifying the model have consequences on the cognitive state of the learner which are not incidental but are in fact desired side-effects. The actual consequences of PLM on the learner's cognitive state have been nevertheless unclear, often described using indefinite words like 'awareness' and 'reflection'.

Two outcomes of PLM on the cognitive state of the learner suggested in this paper are:

1. The acquisition by, and reinforcement in, the learner of declarative representations of problem-solving knowledge.
2. The establishment of *common (mutual) knowledge* between the participants in the learner modelling process.

A description of the cognitive state of the learner is presented in section 2. The two consequences of PLM on the learner's cognitive state outlined above are discussed in sections 3 and 4. Finally, some implications for the structure and content of the learner model under PLM are advanced in section 5.

## 2 The learner's cognitive state

A convenient way of describing the learner's cognitive state is in terms of two types of representation for knowledge: declarative and procedural (Anderson, 1993; Dillenbourg and Self, 1992; Ryle, 1949).

**Declarative representation** A *flexible* way of storing knowledge which can be reported, applied in different contexts, and accessed in several ways for coping with unexpected situations.

**Procedural representation** An *efficient* form of encoding knowledge to be "executed" in performing specific tasks, but extremely difficult for its possessor to articulate.

Classical examples of knowledge represented declaratively include geographical knowledge (for instance, the continent where Mexico is located). Typical examples of knowledge encoded declaratively are the algorithm of multicolumn addition, and our ability to speak a language.

The distinction does not mean a sharp partition of knowledge itself in two categories. Some knowledge might, in fact, be worthy of being represented in both declarative and

procedural form, because that would allow the owner of those representations to reason about such knowledge, modify it, use it in more flexible ways, and also follow it for efficient problem-solving.

The cognitive state of the learner is illustrated graphically in figure 1. The boxes de-

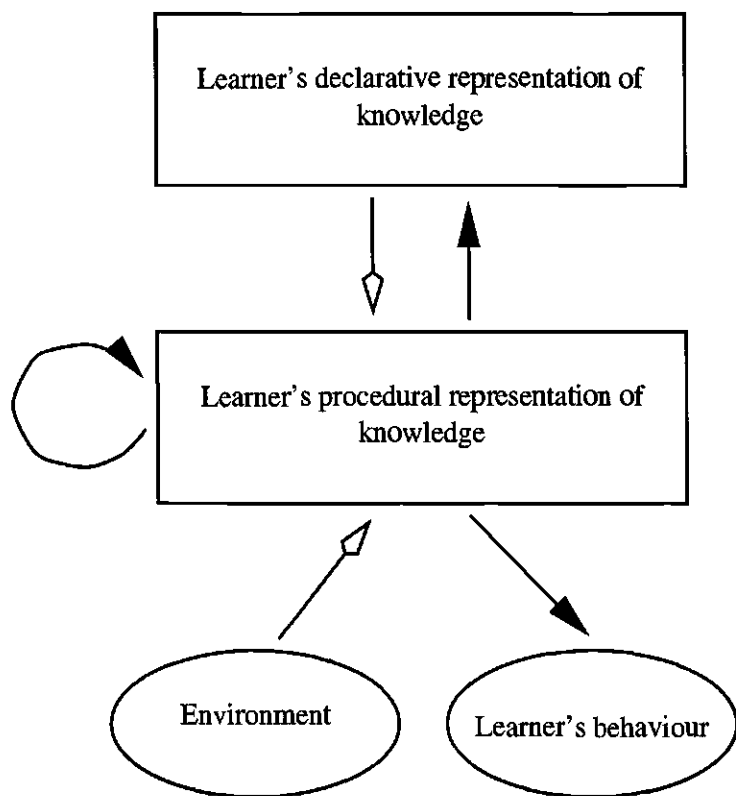


Figure 1: The cognitive state of the learner.

note two repositories of knowledge, each one corresponding to one kind of representation for knowledge; hollow-head arrows indicate input sources, and solid-head arrows indicate output from the execution of knowledge. In execution, procedurally encoded knowledge receives input data from the environment and declaratively represented knowledge, but not procedurally encoded knowledge itself. The results are behaviour and new representations, both declarative and procedural, for knowledge.

The unconscious nature of the learner's procedural encoding of knowledge makes it unlikely that any awareness and reflection promoted by PLM will have any significant direct effect on that representation. The consequences of PLM should then be looked for in the learner's declarative (and conscious) representation for knowledge.

### 3 Acquisition of declarative representations of knowledge

In the following discussion, it will be assumed knowledge can be described as being composed of small identifiable pieces. Some of them have a tendency to be represented declar-

atively, and they are regarded as “conceptual knowledge”; others are usually encoded in a procedural form, and are considered to be of a “problem solving” nature. Besides “valid” units of knowledge, “mal-formed” units will also be taken into account, corresponding to possible “misconceptions” that might be ascribed to the learner. The learner model will consist of a subset of the union of domain knowledge and misconceptions, under appropriate conditions of consistency (Aiello et al., 1991), and it will be assumed that this is a faithful representation of the learner’s knowledge.

When inspecting such a learner model, the learner might find it contains a piece of knowledge the system regards as known by the learner. In cases where that piece of knowledge were rather useful for performing a specific task, it may be that the learner has only a procedural encoding of it: she may be able to perform the task and at the same time unable to articulate her knowledge. For example, the production rules in MR. COLLINS (Bull and Pain, 1995) represent knowledge about how to place object pronouns in European Portuguese sentences. A learner may have a procedural encoding of this knowledge, and thus be able to locate the pronouns in their correct positions, yet she may be incapable of articulating her knowledge due to a lack of a declarative representation of it. In general, it may happen the learner actually knows a piece of knowledge but is unaware of this fact; that is, she does not know that she knows it.

A relatively simple piece of knowledge recorded in the learner model as known by the learner can become part of the learner’s declaratively represented knowledge when she finds it in the learner model. In the example above, a declarative account of the rules for locating object pronouns in European Portuguese sentences can be acquired by the learner through inspecting MR. COLLINS’s learner model, and this fact is interpreted by Bull (1997) as MR. COLLINS having promoted some degree of language awareness. This gain in declarative representations of knowledge is regarded as the most immediate effect of PLM on the learner’s cognitive state.

It is important to clarify that such an acquisition of knowledge will rarely happen in cases where the user is superficially browsing the learner model; rather, the user has to be concentrated in the task. The situation can be encouraged by challenging the learner to negotiate the content of the learner model (Bull and Pain, 1995).

The effect is the same in cases where that piece of knowledge actually corresponds to a misconception: it also gets encoded in the learner’s declarative memory when she discovers it in browsing the learner model. The main difference arises in that this effect may be used for remediation.

## 4 Establishment of common (mutual) knowledge

Following the notation in (Fagin et al., 1996), the fact that a piece of knowledge—either “conceptual” or “problem solving” knowledge—is encoded in the declarative memory of the learner can be represented by the term

$$K_l p, \tag{1}$$

where  $p$  stands for the piece of knowledge and the operator  $K_l$  should be read as ‘it is known by the learner that ...’.

#### 4.1 The case of domain knowledge

If the piece of knowledge denoted by  $p$  is not a misconception of the learner but actually part of the domain knowledge held by the system, then not only  $K_I p$  but also

$$K_s p \quad (2)$$

is the case, where the operator  $K_s$  should in turn be read as ‘it is known by the system that ...’. Hence  $p$  can be regarded as knowledge *shared* between the learner and the system; a fact denoted in (Fagin et al., 1996) by the term

$$E p, \quad (3)$$

where the operator  $E$  is read as ‘everyone knows that ...’.

The fact that a piece of knowledge  $p$  is registered in the learner model as encoded in the learner’s declarative memory means the learner knows it, to the best of the system’s knowledge. That fact is represented by the term

$$K_s K_I p. \quad (4)$$

When the learner inspects the part of the learner model represented by (4), she knows it is looking at a representation of her knowledge held by the system. Thus it may be assumed a reinforcement is given not only to her knowledge of  $p$  but also to her knowledge about her knowledge of  $p$ ; that is, she will know that she knows  $p$ , a fact denoted by the term

$$K_I K_I p. \quad (5)$$

If the learner is allowed to inspect the information recorded in the learner model about the system’s judgement on her knowledge of  $p$ —that is, whether the system considers the learner knowledge of  $p$  “real knowledge” or “misconception”—then in this case it is possible to assume the learner knows that the system knows  $p$ ,

$$K_I K_s p. \quad (6)$$

This fact, together with (4) and (5), will support asserting that not only everyone knows  $p$  but also that *everyone knows that everyone knows  $p$* , a fact denoted by the term

$$E^2 p. \quad (7)$$

This is the case if it is also possible to assume the system knows that it knows  $p$  too, a fact denoted by the term

$$K_s K_s p. \quad (8)$$

The status of this term depends exclusively on the design of the system’s knowledge representation and reasoning capabilities, and not on the inspectability of the learner model.

The importance of (7) is that it makes  $p$  closer to the status of *common knowledge* between the learner and the system, a notion represented in (Fagin et al., 1996) by

$$C p$$

and defined as

$$Cp \text{ if and only if } E^k p \text{ for } k = 1, 2, \dots$$

I believe the leap from  $E^2 p$  to  $Cp$  as an effect of PLM, is justified on the basis of what happens daily in human-human interactions. For example, when Marcio says to Mike 'I have bought two bottles of wine, one for me and another for Paulo', they both knew that the bottles were for a party at Mike's, and that Paulo was a friend who was also going to the party: these two facts were shared knowledge. They also knew that they shared their knowledge of these two facts,  $E^2(f_1 \wedge f_2)$ , and that they knew that they knew that they knew,  $E^3(f_1 \wedge f_2)$ , etc.: the two facts were indeed part of their common knowledge. However, the status of common knowledge does not seem to be reached through a long reasoning process, nor seem to require an infinite model of each other interlocutor, but it relies on basic assumptions about human behaviour in the social environment (c.f. Grice, 1967).

If a system could construct a faithful learner model with the participation of the learner, and present it to the learner for inspection and modification, then this would elevate the system from being a sophisticated tool to something more akin to a partner. This would make feasible the learner's ascribing some form of human-like behaviour to the system, and hence permit the idea of them having common knowledge.

#### 4.2 The case of a misconception

If the piece of knowledge denoted by  $p$  is actually a misconception of the learner, rather than actual domain knowledge, then the term (2) changes to

$$K_s \neg p \tag{9}$$

and is no longer possible to assert  $Ep$ . However, the terms (4),(5) are still valid, making possible to assert instead

$$EK_l p. \tag{10}$$

On the other hand, the terms (6) and (8) change to

$$K_l K_s \neg p \tag{11}$$

and

$$K_s K_s \neg p \tag{12}$$

respectively, making possible to assert

$$EK_s \neg p. \tag{13}$$

An argument similar to the one presented in the previous section would produce as a result that in this case it will be common knowledge that  $p$  is considered a misconception of the learner, in opposition to the actual knowledge held by the system; that is,

$$C(K_l p \wedge K_s \neg p). \tag{14}$$

### 4.3 Logical omniscience and related issues

It has to be taken into account that a learner using an intelligent educational system is far from being a *logically omniscient* agent, knowing all logical truths and all the consequences of her knowledge (Fagin et al., 1996). On the contrary, learners generally have incorrect, incomplete and inconsistent knowledge, do not believe all the logical consequences of their beliefs—at least not in the sense of standard logic—and seem to reason using a different logic from an expert. Having stated these characteristics of learners, it should be noted that they do not invalidate the argument given above, at first it might seem: our argument relies on the inspectability of the learner model by the learner, not on her logical omniscience.

## 5 Modelling the effects of PLM

Enabling an intelligent educational system to model the effect of PLM on the cognitive state of the learner means supplementing the system with the ability to:

1. Model the reasoning of the learner about her own knowledge when inspecting the content of the learner model.
2. Register in the learner model the consequences of that reasoning: declarative representations of knowledge, including beliefs of the learner about her own and the system's beliefs.

Learner models which take into account system and learner beliefs about each other's beliefs have been developed, notably BGP-MS (Kobsa and Pohl, 1995). However, BGP-MS considers neither the system's nor the learner's beliefs about their own beliefs—like the ones expressed by terms 5 and 8 above. The reasons given are the lack of both a clear theory and practical need. Besides that, BGP-MS does not distinguish between any  $E^np$  and  $Cp$ . Yet PLM effects on the cognitive state of the learner could provide just such a practical justification for an extended learner model of the kind outlined in the theory above.

## 6 Conclusions

The consequences of participative learner modelling (PLM) on the cognitive state of the learner have been described as the acquisition of declarative representations of knowledge by the learner, and the establishment of common knowledge between the participants in the modelling task. However, further study of PLM is required, especially with respect to the actual effect PLM has on learners in a tutorial situation.

The learner's and the system's beliefs about their own beliefs, having been neglected in mainstream research on learner modelling, are two main focus of PLM; they demand careful consideration. Without them, the justification and usefulness of any form of PLM—as collaborative learner modelling, or negotiation in learner modelling—are questionable.

## 7 Acknowledgements

We are very grateful for interesting comments and suggestions on ideas contained in this paper by Damjan Bojadziev, Paul Brna, Tom Conlon, Luis Pineda, Alan Smaill, and Henry Thompson. We would also like to thank Anders Bouwer, Shari Trewin, Angel de Vicente, and the anonymous workshop referees for their useful feedback on earlier versions of this paper.

Rafael Morales is being supported by Instituto de Investigaciones Eléctricas and CONACYT, Mexico, under scholarship 64999/111091.

## References

- Aiello, L. C., M. Cialdea, and D. Nardi (1991). Reasoning about student knowledge and reasoning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1087–1093. Sydney, Australia. 3
- Anderson, J. R. (1993). *Rules of the Mind*. Lawrence Erlbaum Associates. 2
- Bull, S. (1997). *Collaborative Student Modelling in Foreign Language Learning*. Ph.D. thesis, University of Edinburgh, Department of Artificial Intelligence. 1, 3
- Bull, S., P. Brna, and H. Pain (1995). Extending the scope of the student model. *User Modeling and User-Adapted Interaction* 5(1), pp. 45–65. 2
- Bull, S. and H. Pain (1995). “Did I say what I think I said, and do you agree with me?” inspecting and questioning the student model. In *Artificial Intelligence in Education, 1995: Proceedings of the Seventh World Conference on Artificial Intelligence in Education*, edited by J. E. Greer, pp. 501–508. Washington, DC, USA. 2, 3
- Cook, R. and J. Kay (1994). The justified user model: A viewable, explained user model. In *Fourth International Conference on User Modeling: Proceedings of the Conference*, pp. 145–150. The MITRE Corporation, Hyannis, Massachusetts, USA. 1
- Cumming, G. and J. A. Self (1991). Learner modelling in collaborative intelligent educational systems. In *Teaching Knowledge and Intelligent Tutoring*, edited by P. Goodyear, chap. 4, pp. 85–104. Ablex, New Jersey. 1
- Dillenbourg, P. and J. Self (1992). A framework for learner modelling. *Interactive Learning Environments* 2(2), pp. 111–137. 2
- Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (1996). *Reasoning About Knowledge*. MIT Press. 4, 4.1, 4.1, 4.3
- Grice, H. P. (1967). Logic and conversation. In *Studies in Syntax*, edited by P. Cole and J. L. Morgan, vol. III. Seminar Press, New York. 4.1
- Kobsa, A. and W. Pohl (1995). The BGP-MS user modeling system. *User Modeling and User-Adapted Interaction* 4(2), pp. 59–106. 5
- Paiva, A., J. A. Self, and R. Hartley (1995). Externalising learner models. In *Artificial Intelligence in Education, 1995: Proceedings of the Seventh World Conference on Artificial Intelligence in Education*, edited by J. E. Greer. Washington, DC, USA. 2
- Ryle, G. (1949). *The Concept of Mind*. Hutchinson, London, England. 2
- Self, J. A. (1988). Bypassing the intractable problem of student modelling. In *Proceedings of ITS’88*, pp. 18–24. Montreal, Canada. 1, 2
- Self, J. A. (1994). *The Role of Student Models in Learning Environments*. AAI/AI-ED Technical Report 94, Computing Department, Lancaster University, Lancaster, UK. 2



Morales, R. and H. Pain (1999). Modelling of novices' control skills with machine learning. In J. Kay, editor, *User Modeling: Proceedings of the Seventh International Conference on User Modeling, UM99*, 159–168. Banff, Canada. Springer Wein New York.

## Modelling of Novices' Control Skills With Machine Learning

Rafael Morales and Helen Pain\*

School of Artificial Intelligence, University of Edinburgh, United Kingdom

**Abstract.** We report an empirical study on the application of machine learning to the modelling of novice controllers' skills in balancing a pole (inverted pendulum) on top of a cart. Results are presented on the predictive power of the models, and the extent to which they were tailored to each controller. The behaviour of the participants in the study and the behaviour of an interpreter executing their models are compared with respect to the amount of time they were able to keep the pole and cart under control, the degree of stability achieved, and the conditions of failure. We discuss the results of the study, the limitations of the methodology in relation to learner modelling, and we point out future directions of research.

### 1 Introduction

Previous research on supporting teaching and learning cognitive tasks has concentrated on high-level skills such as problem-solving in mathematics and physics, programming, and second language learning. Acquisition of real-time control skills of the sort required for playing a musical instrument, driving a vehicle or operating a tool have received much less attention. This paper attempts to make a contribution to the latter, more neglected area, with respect to learner modelling. Descriptions of strategies followed by apprentices of the simple task of balancing a pole attached to a cart are obtained by applying machine learning techniques to traces of the apprentices' behaviour. We consider whether these can be regarded as adequate representations of the evolving control skills of novices.

Machine learning techniques have been applied to pole balancing and other controlling tasks like flying a plane, operating a crane, and production scheduling (see Bratko et al., 1997, for an overview; Michie et al., 1990; Michie and Camacho, 1994; Urbančič and Bratko, 1994). The methodology, termed *behavioural cloning* (Michie et al., 1990), was originally motivated by the difficulties encountered in getting expert controllers to produce detailed explanations of their skills that can be embedded in programs. Learner modelling differs, however, in a number of respects from expert modelling, owing to the fact that the subject is not an expert, but a beginner whose behaviour manifests faulty and inconsistent performances.

The use of machine learning techniques for learner modelling has a long history (e.g. Gilmore and Self, 1988; Langley et al., 1984; Sleeman, 1982; Webb and Kuzmycz, 1996; see also Sison and Shimura, 1988). Machine learning offers the possibility of data-driven learner modelling, focused on the actual behaviour of the learner, without the prerequisite of detailed descriptions

---

\* We thank Tom Conlon, Donald Michie, Kaśka Porayska-Pomsta, Michael Ramscar, Shari Trewin, Angel de Vicente, and three anonymous reviewers for comments on this paper. William Cohen deserves special thanks for allowing free use of RIPPER, and his prompt and kind response to all our questions. Rafael Morales is being supported by CONACYT and the Instituto de Investigaciones Eléctricas, Mexico, under scholarship 64999/111091.

of domain knowledge and its common variations (the latter usually referred to as 'misconceptions', 'bugs', or 'mal-rules'; cf. Sison and Shimura, 1988). The reduction of assumptions about domain knowledge gives grounds for expecting a decrease in the bias of the diagnosis, and hence greater flexibility to accommodate different (human) learning styles and different conceptions of domain knowledge (Jonassen and Grabowski, 1993). However, because machine learning does not necessarily relate to human learning, claims about the psychological status of models constructed with it have varied. Advocates of the approach have either attempted to embed their techniques in broader psychological theories (e.g. Langley et al., 1984), or they have assumed to model solely competence in the task, without claiming to describe plausible cognitive processes of human learners (e.g. Gilmore and Self, 1988; Webb and Kuzmycz, 1996).

Our research differs from related work on behavioural cloning in that it focuses on apprentices, rather than experts. We are interested in making matches between subject and clone behaviour, whereas research on behavioural cloning has focused on maximising the clone's expertise. Our work differs from previous work on using machine learning for learner modelling in the time-constrained and highly dynamic nature of the domain, which demands a different approach to preparing the input data and evaluating the adequacy of the models. We focus on devising methods for diagnosing novice performers of real-time, control-like tasks, constructing representations of their strategies based on traces of their behaviour, and checking that the representations are faithful models of the novices' competence in the task. As to the psychological credibility of the models, we adopt a conservative approach: our intention is not to build accurate psychological models, but rather models that we could offer to learners as abstract representations of the strategies they follow; models that learners can identify themselves with and inspect as part of their learning process. This facet of the present work derives from our ongoing research on *participative learner modelling* (Morales et al., 1998).

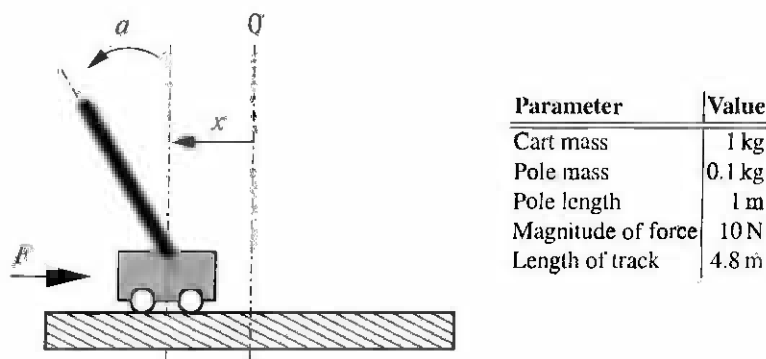
To briefly summarize the rest of this paper, Section 2 describes an empirical study used to gather subject data. Section 3 we describes the behaviour of the participants in the task. The procedure of preparing the traces and inducing the models is presented in Section 4. The predictive power of the models is analysed in Section 5, their individualised nature is discussed in Section 6, and a comparison between the behaviour of participants and their respective models is made in Section 7. The general discussion of results and conclusions are given in Section 8.

## 2 The study

The basic task explored involved balancing a pole (inverted pendulum) attached to the top of a cart (wheeled vehicle) mounted on a straight track of finite length (Figure 1); the pole could fall over the cart only along the vertical plane passing through the track. The whole device could be controlled only by the application (or not) of a force of fixed magnitude, parallel to the track, but with a choice of left or right direction. A simulator based on existing code made available by Finton (1994) was coupled with a graphical user interface and then used instead of a physical device. In the empirical study, every *control run* started with a still pole tilted randomly  $\pm 6$  degrees on a still cart placed in the centre of the track, and ended whenever a *crash* occurred (i.e. any time the cart fell off the end of the track or the pole reached a horizontal position). User input was restricted to pressing arrow keys:  $\uparrow$  to start a control run,  $\leftarrow$  to push the cart to the left, and  $\rightarrow$  to push the cart to the right. User keystrokes were collected for every 100 ms, the action

corresponding to the last keystroke sent to the simulator, and the subsequent new state of the device displayed. The simulator was set up to calculate the state of the device in time increments of 20 ms. The combination of timings, of the interface and the simulator, resulted in a simulation five times slower than the real pole and cart device.

Six subjects took part in the study. They received a brief introduction to the task and the interface, and then were instructed to try keeping the pole in a non-horizontal position and the cart on the track. They were told to start a new control run after every crash. After five minutes of playing with the system, the participants were instructed to continue for another five minutes, and prompted to try harder in pursuing the task.



**Figure 1.** The pole and cart device. A position of the cart on the right (left) half of the track is taken to be positive (negative). An inclination of the pole to the right (left) of the vertical is considered positive (negative).

### 3 Behaviour of the participants

A straightforward measure of the performance of the participants is *control run length*, i.e. the amount of time they were able to avoid a crash. Because a participant could achieve a given control run length in several different ways, exhibiting different “control styles,” we conceived an additional *index of stability* to give a more detailed account of the control process than the raw end result. The control strategy shown in Figure 2 defines a decreasing order of relevance of the state variables for controlling purposes, from the angular velocity of the pole to the position of the cart. Following it every state of the pole and cart was classified into one of five categories, in increasing order of stability: *falling* (0.0025), *tilted* (0.0474), *leaving* (0.5), *displaced* (0.9526), and *stable* (0.9975); the stability index per category was obtained by evaluating the sigmoid function  $s(x) = 1/(1 + e^{-3x})$  at  $x = -2, -1, 0, 1, 2$ . The stability of a control run was then calculated by summing up the stability of all its states, divided by the total of states in it. An overall stability index per participant was calculated as the cumulative effect of the whole set of states of the pole and cart generated by each participant. The last characteristic of the participants’ behaviour we considered

<i>falling:</i>	if $\dot{a} > 0.5$ push right else if $\dot{a} < -0.5$ push left
<i>tilted:</i>	else if $a > 0.07$ push right else if $a < -0.07$ push left
<i>leaving:</i>	else if $\dot{x} > 0.4$ push right else if $\dot{x} < -0.4$ push left
<i>displaced:</i>	else if $x > 0.5$ push right else if $x < -0.5$ push left
<i>stable:</i>	else do nothing

**Figure 2.** Adaptation of a succesful control strategy from (Michie et al., 1990). Angles ( $a$ ) are measured in radians (clockwise direction is positive), angular velocities ( $\dot{a}$ ) in radians per second; cart positions ( $x$ ) in metres, and cart velocities ( $\dot{x}$ ) in metres per second. The adaptation consists in a change of the thresholds for the position of the cart, from zero to  $\pm 0.5$ .

was the object they finally crashed, either the pole or the cart. Statistics per participant of these three aspects of their behaviour are presented in Table 1.

The behaviour exhibited by the group was far from the expert behaviour reported by Michie et al. (1990), whose expert was able to control the device for five minutes. In our study the longest control run lasted only two minutes, and even that was atypical of the performance of the participants. According to control run length, the participants seem to split into three categories: “short run” performers (participants  $S_3$  and  $S_6$ ), “medium run” performers (participants  $S_2$  and  $S_4$ ), and “long run” performers (participants  $S_1$  and  $S_5$ ). The lack of expertise and the variety in the group of participants can be regarded as a useful test of the robustness of the methodology and the adequacy of the models it produces.

Although it was possible to achieve long runs within a adventurous style (low stability), and to get short runs in a cautious manner, we expected a positive correlation between the index of stability and control run length. The results on overall stability, shown in Table 1, indicate that participants  $S_3$  and  $S_6$  had great difficulties at controlling the angular velocity of the pole; they achieved a very small number of stable and displaced states.  $S_1$ ,  $S_2$ , and  $S_4$  performed better, achieving higher stability scores, but below the outcome of  $S_5$ . In general, there was high variability in stability across control runs for all participants, although with a tendency to gain in stability over time. The performance of participant  $S_6$  showed the least variation.

To prevent miscounting as a cart crash a loss of control of the pole from which it is impossible to recover even if there were more space in the track, a pole with an inclination of more than twelve degrees in either direction was regarded as crashed, as in (Bratko, 1995; Michie et al., 1990). As before, the behaviours of participants  $S_3$  and  $S_6$  were quite similar: both had difficulties in controlling the pole. Participants  $S_2$  and  $S_4$  again had similar behaviour, exhibiting less difficulties in controlling the pole than  $S_3$  and  $S_6$ . Participant  $S_1$  achieved relatively good control over the pole early in the study, but did not improve very much afterwards. On the other hand, participant  $S_5$  had initial difficulties at controlling the pole, followed by a dramatic improvement.

## 4 Modelling procedure

All user actions on the simulated pole and cart were recorded in trace files in the general form *device status*  $\rightarrow$  *user action*. The *device status* contained the values of the pole and cart positions and velocities, as displayed on the screen for the last 100ms. The *user action* was either the

**Table 1.** Statistics per participant of control run length, stability index, and crashing conditions. Medians and geometric means are included because the distributions are skewed positively. Runs lasting less than 2.5 seconds were not taken into account in calculating the statistics of control run length and stability. There were five such control runs, and three of them are counted in the *Other* category in the section on crashing conditions

Property	Statistics	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
Control run length (in seconds)	Median	23.1	16.7	9.4	18.6	19.8	9.4
	Geom. mean	22.6	17.1	9.4	18.5	20.3	9.7
	Mean	32.0	20.7	10.8	21.6	31.9	10.5
	Std dev.	29.1	13.3	5.9	12.9	29.2	5.0
Overall stability index		0.25	0.30	0.10	0.27	0.47	0.03
Crashing conditions	Pole	12	22	48	19	13	56
	Cart	7	8	0	9	5	2
	Other	0	0	1	0	4	0

action corresponding to the user’s last keystroke in the last 100 ms, or a “no action” encoding the lack of a user keystroke in the same period. The procedure for extracting the models consisted of three steps: preparation of the traces for diagnosis, induction of a set of production rules, and informed refinement of it into a learner model.

Two problems had to be solved in the preparation stage. Due to possible delays between perception and action, we could not simply assume that the user action stored in a record corresponded to the pole and cart status in that same record: it could correspond to an earlier status of the device, displayed some hundreds of milliseconds before and hence stored in a previous record. A related problem was the treatment of no-actions, introduced by the system every time no keypress occurred for the last 100ms; the greater the reaction delay, the more *undesired no-actions* it caused. Observations during the study made clear that some no-actions were the correct interpretation of the participants’ intentions, and hence it would be unwise for us simply to remove all no-actions from the traces.

The mean of the lag between the start of a control run and the issue of the participants’ first action provides an estimated upper limit to reaction time in the task ( $N = 202$ , mean = 706.4 ms, std = 318.0ms; median = 647.5 ms). It is likely for the task to become less dependent on raw reaction time after the first action has been issued<sup>1</sup>. We chose the value of 300 ms for reaction time, based on the mean lag of 343.2 ms between pairs of consecutive actions ( $N = 10022$ , std = 442.3 ms; median = 180.0ms), remarkably close to the estimated reaction time of 350 ms to pressing a key in response to a simple visual stimulus produced on the screen (Cotterill, 1989, cited by Michie et al., 1990). In practice, that meant aligning the device state and user actions with a shift of three,  $device\ status_k \rightarrow user\ action_{k+3}$ , and stripping the traces of all sequences of no-actions with less than three elements (cf. Michie et al., 1990).

<sup>1</sup> Decisions and actions, even if elicited in response to the present state of the device, are primed by previous states and actions; accumulated knowledge of the task allows some actions to be planned in advance; and some degree of parallelism of the cognitive processes of states perception, selection of responses, and execution of motor actions evolves.

We divided the sequence of control runs of each participant up in overlapping sections of roughly five minutes long (such that they did not split any control run). A five minute window was displaced over the sequence of control runs in steps of around thirty seconds, resulting in six sections for all participants apart from  $S_1$ , who got only four sections. Variations in the number of sections and their span came as a result of not splitting control runs and the variability in the length of control runs achieved by the participants. The groups of records resulting from the alignment, filtering, and sorting out described above were finally presented as input data to RIPPER, a domain-independent rule-learning system (Cohen, 1995).

Specific traits of the domain, such as symmetry, the range of the variables, and their interrelationship, could not be dealt with in the induction process itself. The limited amount of data, and the fact that both the starting and final states in every control run are necessarily asymmetrical, obscured the symmetry of the domain. In order to compensate for these limitations, we introduced symmetrical cases as input to the induction process: for every case *device status*  $\rightarrow$  *user action*, a new case with a symmetrical device status and opposite action was included too. Post-processing of rule sets consisted of substituting a default rule for all the rules issuing no-actions.

## 5 Predictive power of the models

The procedure described in the previous section produced a number of rule sets per participant—four from  $S_1$ , and six from all other participants. For these rule sets to be properly called *individualised* models, they should exhibit two properties: first, they should match the subjects they were extracted from, and second, they should differ from each other<sup>2</sup>.

For each five minute window (as described above) a model was induced. The predictive power of this model was tested on the following (roughly) thirty second window of control runs (again avoiding splitting individual runs). The final five minute window in each case was not verified. The results, presented in Table 2, were error rates between 8.4 and 44.2 percent ( $N = 28$ , mean = 30.0, std = 8.7). There is a significant difference between lower error rates for participant  $S_6$ , and higher error rates for  $S_2$ ,  $S_3$ ,  $S_4$ , and  $S_5$  (One-way analysis of variance:  $dfs = 4, 20, 24$ ;  $F = 7.1407$ ,  $p < 0.001$ . Tukey-HSD test with significance level 0.05. Results on participant  $S_1$  are excluded from the tests because of their different number of models).

Table 2. Predictive power of the models per participant.

Property	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
Mean no. of cases	366.7	154.0	167.0	213.8	235.2	207.2
Min. error (%)	27.0	26.8	21.7	28.5	23.1	8.4
Max. error (%)	36.4	39.9	42.3	44.2	41.7	21.6

Although these results are statistically highly significant when compared to raw random guessing (the binomial test of the combination of the least number of cases,  $N = 118$ , and the

<sup>2</sup> If the models were all very similar, they could still be individualised models, but we would not have evidence supporting that.

worst error rate gives  $p < 0.001$ ), they do not argue for a good match by themselves. It could be argued instead that an undetermined amount of the error rates is due to errors in the alignment of states and actions (Section 4). Despite this caveat, it is worth mentioning that a mean of 72.4% of actions per participant were predicted by their models (min = 49.4, max = 93.5, std = 11.9), and that only a mean of 5.3% of actions per participant were predicted in the wrong direction (min = 0, max = 14.5, std = 4.8).

## 6 Differentiability of the models

Because participants in the study exhibited clear differences of behaviour in their attempts to control the pole and cart, we expected such differences to be apparent also in their models; i.e. models from the same participant should be similar among themselves and different to those from other participants. We opted for a simple dissimilarity measure between models: the level of disagreement in their predictions. A straightforward measure was defined in terms of the traces of the participants' behaviour as

$$d(M_a, M_b) = \frac{1}{\#C_a + \#C_b} \left( \sum_{c \in C_a} (M_b(c) - C_a(c))^2 + \sum_{c \in C_b} (M_a(c) - C_b(c))^2 \right), \quad (1)$$

where  $M_a$  and  $M_b$  are models;  $C_a$  and  $C_b$  are case sets from which  $M_a$  and  $M_b$  were extracted, respectively;  $C_x(c)$  is the action corresponding to case  $c$  in  $C_x$ ;  $M_x(c)$  is the action predicted by model  $M_x$  for case  $c$ ; and actions are encoded as -1 for pushing-left, 1 for pushing-right, and 0 for no-action. The problem with this measure is that it depends on the accuracy of the alignment between states and actions, as recorded in the case sets. A second measure allows to compare directly the predictions given by models on the basis of a sample of the set of states generated by the participants during the study. It is defined as

$$d(M_a, M_b) = \frac{1}{1000} \sum_{s \in S} (M_a(s) - M_b(s))^2, \quad (2)$$

where  $S$  is a sample of one thousand of such states.

Two cluster analyses were then applied to both dissimilarity matrices, using average group and Ward's method (Everitt, 1993). The number of groups were selected on the basis of visual inspection of the dendrograms produced by the clustering methods and plots generated by multidimensional scaling. The analyses suggested three and four groups using dissimilarity measure (1), and five groups using dissimilarity measure (2). Overall, they identify models corresponding to participants  $S_3$  and  $S_6$ ; Ward's methods also distinguished (some) models corresponding to  $S_1$  and  $S_2$ ; but none of the analyses distinguished between models of  $S_4$  and  $S_5$ . The analyses agreed among them in 405 of 561 decisions (72%), and agreed with the known grouping in 360 of the decisions (64%)—there were  $\binom{34}{2} = 561$  pairs of models that could be classified either in the same or different group. The best match with the known grouping was given by the Ward's method using dissimilarity measure (2): 480 of 561 decisions (86%). A binomial test shows that all results are highly significant in reproducing the correspondence between models and participants (random guess with  $N = 561$  and  $k = 360$  gives  $p \ll 0.0001$ ).

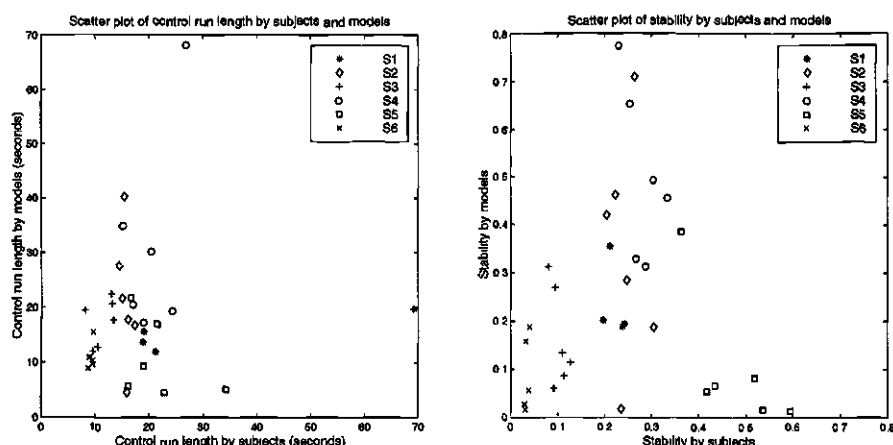


Figure 3. Comparison of control run length and stability achieved by subjects and models. For clarity, control run length for model  $M_{4,6}$  is not shown (235.06 seconds).

## 7 Behaviour of the models

We built an interpreter for the models with a fixed reaction time of 300 ms, and ran every model fifty times, with the starting state of the pole and cart as before. The geometric mean of control run length and overall stability were measured both in the control runs produced by executing each model and in the control runs the model was derived from in the first place (Figure 3). The comparison between subjects' and models' performance gave three general results:

1. Models of participants  $S_3$  and  $S_6$  exhibited behaviour very similar to their respective subjects: short control runs, low stability, and difficulties in controlling the pole. Models of  $S_6$  also showed high predictive power.
2. Models of participant  $S_4$  usually outperformed participant  $S_4$ , with long control runs of high stability and good control. This relates to the *clean-up effect* observed by Michie et al. (1990), consisting of the behavioural clone outperforming the expert it was derived from.
3. Models of participant  $S_5$  often performed considerably worse than participant  $S_5$ , and on several occasions performed worse than models of  $S_3$  and  $S_6$ , despite the fact that  $S_5$  was far better at controlling the pole and cart.

## 8 Discussion and conclusions

The assumption of a fixed amount of reaction time, both within a participant's control runs and among participants, made the alignment of states and actions in traces of the participants' behaviour easier, and we did not have any good arguments for doing otherwise. However, this neglects differences in sensorimotor abilities among participants, and the possibility of improvement with practice. It is possible that some participants overcame their sensorimotor limitations



by taking into account additional information for short term planning, and issued actions more accurately.

Poorer control of the pole and cart produces a good sample of the whole space of possible states of the device, and corresponding control actions, from which accurate models can be induced. Behavioural cloning produces a small cleanup from low quality control and improves considerably medium quality control. In contrast, tight control of the pole and cart produces a biased sample of the whole space of control from which it is more difficult to induce a complete model of the control strategy followed by the subject. This, combined with the clean up attempted by behavioural cloning, resulted in brittle, low quality control strategies. Furthermore, participant  $S_5$  had the most economical control strategy, in terms of the number of actions issued. That property translated into traces containing lots of *no-actions* that were overestimated by the machine learning algorithm<sup>3</sup>, which in turn produced extremely economical models that were unable to keep the pole and cart in conditions such as those contained in the sample of the control space from which the models were induced.

Some issues concerning the application of machine learning to learner modelling, as performed in our study, need to be considered. Very little knowledge about the domain was taken into account for the production of the models. More information about the symmetry of the domain, the nature of the actions, and critical regions in the space of control during the process of induction, would produce better results. Also, the resulting models are "flat," embodying a purely reactive conception of the task. However, participants were presented with a goal: keep the pole and cart under control for as much time as possible; at least some of them appeared behaving in a goal-oriented way. In this respect, other machine learning approaches, like inductive logic programming, could be interesting alternatives to the more traditional machine learning techniques employed in our study (cf. Chiu et al., 1997). An approach to learner modelling simply based on extraction of the current model from the last minutes of the learner's behaviour appears to be too limiting, resulting in high variability of the models over time. A proper learner model maintenance module, comprising machine learning techniques as subcomponents, would be needed. Ten minutes of controlling behaviour per participant provides too little data to say anything conclusive about the learning of the participants and the evolution of their models.

In conclusion, we have presented an empirical study in applying machine learning to model novices in the task of controlling a pole on a cart. Our results indicate it succeeded in distinguishing between different subjects, hence producing clearly different models for them. A static test on data collected from the participants showed the models performed well at predicting the actions of the subjects in the short term, especially if the likely presence of noise due to our assumption of fixed reaction time is taken into account. Dynamic tests of the models consisted of executing them and comparing their performance with the original performance of the participants. A close correlation was noticed for three of the participants, while an evident discrepancy was discernible in the remaining cases. We advanced an explanation for the latter failure in terms of reaction time, the clean up of control behaviour and the high degree of stability achieved by some participants, and suggested some ways to overcome the deficiencies of our approach. Our results show machine learning can be a useful tool for diagnosis in learner modelling in domains

---

<sup>3</sup> We tried to ameliorate this effect by weighting false positives and false negatives in RIPPER. The effect was clearly appreciated in better performance of models from  $S_2$ .

involving control tasks, although it needs to be enhanced with more domain-specific knowledge, and embedded into a more comprehensive learner model maintenance system.

## References

- Bratko, I., Urbančič, T., and Sammut, C. (1997). Behavioural cloning of control skill. In Michalski, R. S., Bratko, I., and Kubat, M., eds., *Machine Learning and Data Mining: Methods and Applications*. John Wiley & Sons. chapter 14, 335–351.
- Bratko, I. (1995). Derivating qualitative control for dynamic systems. In Furukawa, K., Michie, D., and Muggleton, S., eds., *Machine Intelligence*, volume 14. Oxford: Clarendon Press. 367–386.
- Chiu, B. C., Webb, G. I., and Kuzmycz, M. (1997). A comparison of first-order and zeroth-order induction for Input-Output Agent Modelling. In Jameson, A., Paris, C., and Tasso, C., eds., *User Modeling: Proceedings of the Sixth International Conference, UM97*, 347–358. Chia Laguna, Sardinia, Italy: Springer Wien New York.
- Cohen, W. W. (1995). Fast effective rule induction. In Prieditis, A., and Russell, S., eds., *Machine Learning: Proceedings of the Twelfth International Conference*. Tahoe City, CA: Morgan Kaufmann.
- Cotterill, R. (1989). *No Ghost in the Machine*. London: Heinemann.
- Everitt, B. (1993). *Cluster Analysis*. London: Edward Arnold, 3 edition.
- Finton, D. J. (1994). Controller-less driver for the cart-pole problem. Available on the World Wide Web at <http://www.cs.wisc.edu/~finton/poledriver.html>.
- Gilmore, D., and Self, J. (1988). The application of machine learning to intelligent tutoring systems. In Self, J., ed., *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instruction*. London: Chapman and Hall Computing. chapter 1, 179–196.
- Jonassen, D. H., and Grabowski, B. L. (1993). *Handbook of Individual Differences, Learning, and Instruction*. Lawrence Erlbaum Associates.
- Langley, P., Ohlsson, S., and Sage, S. (1984). A machine learning approach to student modeling. Technical Report CMU-RI-TR-84-7, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.
- Michie, D., and Camacho, R. (1994). Building symbolic representations of intuitive real-time skills from performance data. In Furukawa, K., Michie, D., and Muggleton, S., eds., *Machine Intelligence*, volume 13. Oxford: Clarendon Press. 385–418.
- Michie, D., Bain, M., and Hayes-Michie, J. (1990). Cognitive models from subcognitive skills. In McGhee, J., Grimble, M. J., and Mowforth, P., eds., *Knowledge-Based Systems for Industrial Control*. London: Peter Peregrinus. chapter 5, 71–99.
- Morales, R., Ramscar, M., and Pain, H. (1998). Cognitive effects of participative learner modelling. In Ayala, G., ed., *Proceedings of the Current Trends and Applications of Artificial Intelligence in Education Workshop*, 49–56. Mexico City, Mexico: ITESM.
- Quinlan, R. J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Sison, R., and Shimura, M. (1988). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education* 9:128–158.
- Sleeman, D. H. (1982). Inferring (mal) rules from pupil's protocols. In *ECAI-82 : 1982 European Conference on Artificial Intelligence*, 160–164.
- Urbančič, T., and Bratko, I. (1994). Reconstructing human skill with machine learning. In Cohn, A. G., ed., *ECAI 94: 11th European Conference on Artificial Intelligence*, 498–502. Amsterdam, The Netherlands: John Wiley & Sons.
- Webb, G. I., and Kuzmycz, M. (1996). Feature based modelling: A methodology for producing coherent, consistent, dynamically changing models of agents' competencies. *User Modeling and User-Adapted Interaction* 5:117–150.

# Understandable Learner Models for a Sensorimotor Control Task

Rafael Morales<sup>1,\*</sup>, Helen Pain<sup>1</sup>, and Tom Conlon<sup>2</sup>

<sup>1</sup> Division of Informatics, University of Edinburgh,  
80 South Bridge, Edinburgh EH1 1HN, UK.  
{R.Morales,H.Pain}@ed.ac.uk

<sup>2</sup> Moray House Institute of Education, University of Edinburgh,  
Holyrood Road, Edinburgh EH8 8AQ, UK.  
tomc@education.ed.ac.uk

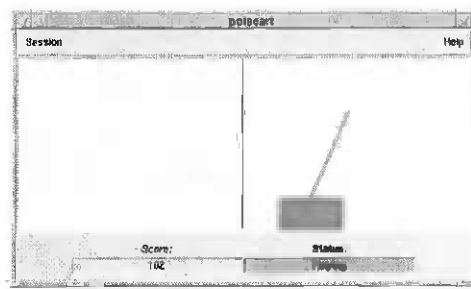
**Abstract.** We discuss the implications of making learner models that can be inspected by learners within the context of a sensorimotor control task—that of balancing a pole hinged to a cart. We argue that the requirement of producing models that are comprehensible by learners limits the options of modelling strategy, constrains model structure and calls for further refinement of model contents. We discuss the issues of modularity of model contents, modality and interactivity of model presentation, and present results from a preliminary evaluation of a graphical interface to learner models for pole balancing.

## 1 Introduction

Inspectability of learner models as a design goal has been advocated for imposing beneficial constraints on learner modelling, like avoiding crude classifications of learners; for encouraging accountability, understandability, and acceptability of learner models; and for adopting a learner-centred perspective [6, 11, 23]. This paper presents a case study of building learner models that can be shown to learners, and understood. The task used as illustrative domain is balancing a pole hinged to a cart; a domain rather different to those inspectable learner models have been previously built for, such as second language acquisition [2, 6, 7, 8], use of a text editor [11], application of engineering procedures [5] and algebraic problem solving [21]. Pole balancing is a sensorimotor control task that has been heavily used as a test domain for machine learning techniques—e.g. [4, 18]. It is appealing because it is relatively simple and facilitates quick construction of fairly individualised learner models [20].

We evaluate the feasibility of making learner models in this domain inspectable by the learners. Careful selection of a modelling strategy is required for learner models to be presentable to learners in a way that facilitates their understanding, and further refinement of model contents may be necessary to improve model comprehensibility. Next section outlines the task of pole balancing and summarises our approach to building learner models in this domain, and in Sections 3, 4 and 5 we discuss issues of modularity of model contents, modality and interactivity of model presentation. Section 6

\* Supported by CONACYT and the Instituto de Investigaciones Eléctricas, Mexico, under scholarship 64999/111091.



**Fig. 1.** Graphical user interface to the pole and cart. A position of the cart on the right (left) half of the track/window is taken to be positive (negative). An inclination of the pole to the right (left) of the vertical is considered positive (negative).

describes an informative study carried out to assess the comprehensibility of a graphical interface to the learner models. Section 7 comes back to the potential benefits of building inspectable learner models and discusses some obstacles to making them more understandable to learners. Finally, Section 8 presents our conclusions.

## 2 From Traces of Behaviour to Learner Models

The task of pole balancing involves controlling a simulation of a rigid pole hinged to a cart that in turn is mounted on a straight rail of finite length (Fig. 1). The device can be controlled only by applying (or not) a force to the cart of fixed magnitude and parallel to the rail but with a choice of left or right direction. User input is restricted to pressing arrow keys:  $\uparrow$  to start a new control trial,  $\leftarrow$  to push the cart to the left, and  $\rightarrow$  to push the cart to the right. The system saves a trace of every control trial per learner containing records of the form (*pole and cart state, action*) which associate each user action in the control run with a state of the pole and cart. After merging several control trials per learner, while taking into account factors related to learners (e.g. reaction time) and the task (e.g. symmetry of the pole and cart system), a final set of records is produced that better describes the correspondence between states of the pole and cart and user actions [20]. These records are the raw material for generating a model of the strategic knowledge employed by each learner when performing the task.

Constructing a learner model from a collection of input-output records can be seen as a classification task. From this perspective, the application to learner modelling of machine learning techniques for classification looks quite attractive and straightforward (yet cf. [24]), with a number of alternatives available: rule discovery, decision trees, Bayesian and neural networks, case-based reasoning, etcetera [19]. Every technique will impose a particular structure upon the resulting model, and the criterion for selecting a specific technique in this case is the production of models structured in a way that facilitates their inspection, understanding, and possible modification by apprentices of pole balancing who are mostly unfamiliar with knowledge representation techniques.

left 960 317 if  $a \leq -0.110277$  .  
 left 789 263 if  $\dot{a} \leq -0.13729$   $a \leq 0.047092$  .  
 right 357 20 if  $x \leq 0.869831$   $a \geq 0.286856$  .  
 right 693 74 if  $\dot{a} \geq -0.218725$   $a \geq 0.151792$   $x \leq 1.42121$  .  
 right 179 25 if  $a \geq 0.104776$   $\dot{a} \geq -0.725516$   $\dot{a} \leq 0.325668$   $x \geq 0.356233$   $\dot{x} \leq 2.38745$  .  
 right 566 143 if  $a \geq -0.055733$   $\dot{a} \geq 0.141926$   $x \leq 0.534309$  .

**Fig. 2.** Angles ( $a$ ) are measured in radians (clockwise direction is positive), angular velocities ( $\dot{a}$ ) in radians per second, cart positions ( $x$ ) in metres, and cart velocities ( $\dot{x}$ ) in metres per second. The two integers in between each rule's action and preconditions are the number of cases correctly and incorrectly classified by that rule, respectively.

Morales and Pain [20] chose production rules because they are easy to interpret in operational terms, have a symbolic character<sup>1</sup> and simple structure that resembles familiar rules of thumb, and support modularity of representation [3, 16]. Furthermore, production rules have been used effectively in modelling human skill acquisition and performance [1]. In an empirical study involving thirty subjects, supervised rule induction with RIPPER [9] produced learner models containing between four and sixteen rules ( $\bar{x} = 10.9$ ,  $s = 3.4$ ), with the number of preconditions per rule varying from two to four ( $\bar{x} = 2.6$ ,  $s = 0.5$ ). An example of a learner model from that study is given in Fig. 2.

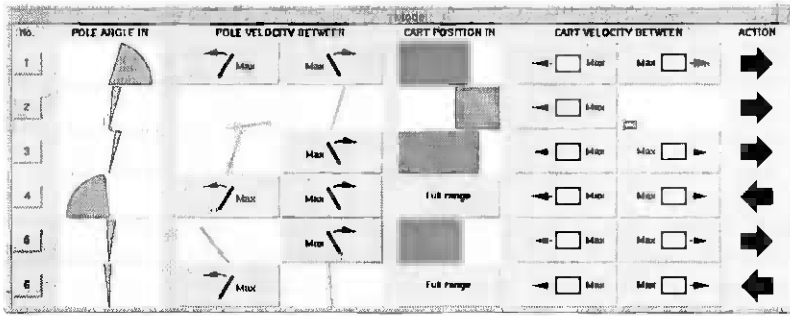
Models like this provide useful information for a number of purposes: to predict which direction the learner will push, if any; to simulate the behaviour of the learner; and to discover similarities and differences with other learner's strategy through statistically comparing their models' predictions [20]. Combinations of positions and velocities where the learner's reaction is markedly different from that of an expert can be discovered by comparing the model to a set of rules representative of expertise. The common feature among these different uses of a learner model is, nevertheless, that they do not require for it to be understandable by the learner. In fact, they do not require human intervention at all, since all is needed is for the computer system to be able to process the model. Things can be quite different when it comes to human processing of the models though, particularly when these humans are the learners being modelled.

### 3 Presenting Models in the Right Modality

Rule preconditions in Fig. 2 are expressed in numerical terms. Although they are very accurate, they are also quite difficult to correlate to the behaviour of the pole and cart on the screen. A different choice of units—for example, degrees instead of radians, or millimetres instead of metres—does not alleviate the problem, because it is not a problem of units but one of *modality*<sup>2</sup>: the lack of consistency in the way the task (Fig. 1) and the learner model (Fig. 2) are presented makes the latter difficult to comprehend despite experience with the former.

<sup>1</sup> Greer et al. [15], on the other hand, present an interesting example of how learner models based on Bayesian networks can be opened to learner inspection.

<sup>2</sup> We use the term *modality* to imply a combination of language and medium, without attempting to give a more precise definition of it. For a deeper discussion of modality, see [22].



**Fig. 3.** Graphical presentation of a refined version of the learner model shown in 2, with preconditions of rules mutually exclusive (Section 4).

A different approach to presenting the learner model is illustrated in Fig. 3. Here the model is presented graphically, in a table-like format in which every row represents a rule; actions are represented by arrows, resembling the arrow keys used to command actions in the interface to the simulator; and rule preconditions are represented by arcs, boxes, and animations of the pole and cart moving with different velocities (every pair of animations denoting the range of velocities defined by the velocities of the animations)<sup>3</sup>. It can be seen that task and model are presented now in the same (graphical) modality. Consequently, much less translation is necessary, the cognitive load on the learner being reduced in this way, and previous experience with the task should better support their understanding of the model.

The issue of consistency between the modalities in which the tasks in the domain and the learner models are presented have not been raised explicitly in previous research on inspectable learner models, where matching of modalities for domain and model presentation varies. In cases of second language learning as the application domain, written natural language has been the modality of choice both for presenting the tasks and the models [2, 6, 8]. In a similar way, de Buen et al. [5] employ mathematical notation, specialised technical terminology and natural language to convey engineering concepts and procedural steps, both during task and model presentation. In contrast, Cook and Kay [11] present their models of users of the SAM text editor using a mixture of text and diagrams (conceptual trees) that differs from the textual interface of the editor itself. Paiva et al. [21] describe language selection as a major difficulty for externalising learner models, although the early example they provide shows the learner model in a logic-based language that is closer to the model's internal representation than to the language in which the task (simplification of algebraic equations) is presented.

The relevance of matching the modalities in which task and model are presented may vary among domains. Sharing modality may be more important in domains that involve the acquisition of sensorimotor skills, like pole balancing, than in domains with

<sup>3</sup> The graphical presentation contains also textual and iconic representations, which are introduced to improve the clarity of the presentation and to diminish the overwhelming effect of too many fast animations.

demands on higher cognitive abilities. In the latter cases, it may be more important for task and model to share more structural properties of their representations, or even a deeper but recognisable set of abstractions, without regards for superficial details of presentation. Individual learners have different preferences of modality and different degrees of representational competence [10, 12] which should be recognised as well.

## 4 Increasing the Modularity of Models

Rules in Fig. 2 are presented in order of importance; i.e. a rule is supposed to fire only on states of the pole and cart that satisfy its preconditions but do not satisfy the preconditions of any rule above it. This ordering has the potential benefits of reducing the number of rules in the model and simplifying their preconditions. However, rule firing is not determined entirely by each rule's preconditions but also by the preconditions of all other rules with higher priority (as if every rule has the negated preconditions of all previous rules). That makes the role of every rule but the first one harder to comprehend.

Fig. 3 contains a refined version of the model presented in Fig. 2, with the preconditions of all rules modified to make them mutually exclusive. Both models are equivalent, yet each rule in Fig. 3 can be understood independently of any other rule. Hence the new model profits more from the modularity of production rules [3, 16].

The issue of modularity has not been explicitly discussed in the context of previous research on inspectable learner models. Although distinct components that may favour a modular design can be observed in all models and their presentations—grammatical rules, description of general tendencies, answers to exercises and their evaluation, specific and more general comments, communicative goals and capabilities, knowledge components, user properties, beliefs and reasoning rules—their use does not intrinsically guarantee modularity, as our example using production rules clearly illustrates.

## 5 Providing Interactivity

Giving learners improved access to their models (as above) does not guarantee that they will pay any attention to their models nor understand them. Learners should feel they are able to interact with their models, through means of exerting influence over their construction, content or structure, as opposed to merely observing them. Most researchers dealing with participative learner modelling have provided means for learners to interact with their models, varying from browsing and direct editing [2, 5, 6, 11, 21], to mechanisms for discussing and negotiating their contents [8, 13].

Our system gives learners direct control over the content of their models by enriching their presentation with editing facilities. Learners can modify the preconditions of any rule either by directly manipulating their graphical presentation or by dragging sliders on a graphical scale; change the resulting action of every rule as they wish, by clicking a button on the mouse; and add, delete, and alter the order of rules in their models (reordering is still important because independence among rules is not guaranteed after learners' editing of their models).

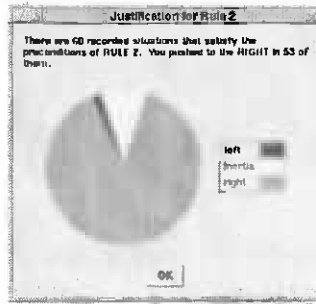


Fig. 4. Justification of a rule.

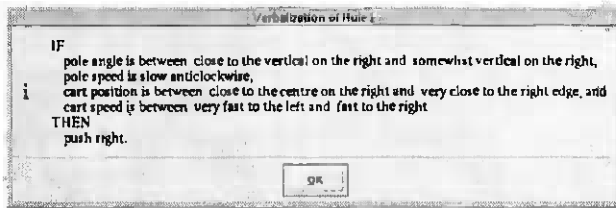


Fig. 5. Example of rule verbalisation. Learners are warned that they should regard the textual description of a rule as a complement to, but not a substitute for the graphical description.

Information about the number of cases correctly and incorrectly classified by each rule is stored in the learner model and can be displayed on request to justify the existence and relevance of every rule (Fig. 4). This simple move makes the accountability of the models more apparent to learners, so it is expected for it to increase model acceptability. Learners can also ask for a short execution of the learner model with initial conditions that trigger a chosen rule, as well as for an explanation of the rule in natural language, the latter being an *if... then* template filled with translations of the rule's preconditions and action (Fig. 5). This facilities for justification, execution and verbalisation of rules offer the possibility of immediate feedback to learners on changes to their models.

## 6 Informative Study

The ease of understanding such a graphical interface was evaluated. Eleven postgraduate students took part in the study, which consisted of three stages. Firstly, each participant was requested to provide some basic background information via a short questionnaire. The second stage consisted in playing with the simulator of the pole and cart for about nine minutes. The third stage required reviewing the graphical presentation of a fictitious strategy for controlling the pole and cart (Fig. 6) and answering a questionnaire about it. Instructions and questionnaires were handed out in printed form to the



No.	POLE ANGLE IN	POLE VELOCITY BETWEEN	CART POSITION IN	CART VELOCITY BETWEEN	ACTION
1					WAIT
2					← Max
3					← Max
4					← Max
5					← Max
6					← Max
7					← Max
8					← Max
9					← Max

Fig. 6. The fictitious strategy for controlling the pole and cart used in the study.

participants at the beginning of each stage and, whereas human intervention was kept to a minimum, the participants were allowed to ask for clarification of any aspect of the program and printed material. They were given as much time as they wished to answer the questionnaires, and they spent between 35 and 75 minutes reviewing the fictitious strategy and answering questions about it.

The questions in the second questionnaire were organised in increasing order of difficulty, from simple questions that tested understanding of the arrow-notation for actions to a final question aimed at testing understanding of the strategy as a whole. Every completed questionnaire was evaluated by the first author, who wrote a model answer for each question, which he then compared to each participant's response. For example, the model answer for the question 'Try to provide a description, in your own words, of the overall strategy defined by the rule set' (Q14) was

The strategy is a "natural" one, with a touch of laziness. That means doing nothing if both pole and cart are centred and moving slowly; pushing in the direction of pole falling if it accelerates, but not if the cart is moving fast in the same direction. In general, the idea is to try to revert to a centred position as long as the cart is not too close to an edge (with the pole falling rather quickly towards it) or the pole is not falling too quickly (the cart being anywhere).

An example of an answer taken as correct is

If the pole is well balanced and the cart is moving slowly in the centre of the screen, do nothing. If the pole is pointing off to the left, is not falling rapidly to the right and the cart is not on the far left, push left. If the pole is pointing to the right and the cart is not on the far right push right. If the pole is upright falling right and the cart is not travelling fast right, push right. Similarly if it is falling left, not going fast left, push left. Otherwise do nothing. (Participant 6)

The results, summarised in Table 1, suggest that question difficulty increased as expected. The questions were weighted according to their number of right answers<sup>4</sup>, and scores calculated for each participant by summing the weights of all correct answers and penalising for incorrect ones. It can be seen that three participants scored less than 50% (2.92), four between 50% and 75% (4.38), and four over 75%—the maximum possible score being 5.85 (100%).

**Table 1.** Answers to the questions about the graphical presentation of a strategy for controlling the pole on a cart. A period (.) indicates a correct answer; 'X' indicates an incorrect answer; '?' indicates an answer that could be regarded as partially (in)correct; and 'N' indicates no answer.

Question		Participants										
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
On actions	Q1	.	X	.	.	.	.	.	.	.	.	.
	Q2	.	X	.	.	.	.	.	.	.	.	.
On cart position	Q3	.	X	X	.	.	.	.	.	.	.	.
	Q4	?	X	X	.	.	.	.	?	?	.	.
On pole angle	Q5	.	X	?	.	.	.	.	?	?	.	X
	Q6	.	X	N	.	.	.	?	X	?	.	.
On pole velocity	Q7	?	X	X	?	.	?	X	?	?	X	.
	Q8	?	X	X	.	.	?	.	?	?	?	?
On cart velocity	Q9	?	X	X	.	.	.	?	?	?	X	.
	Q10	.	X	?	.	.	.	.	.	?	?	.
On whole rules	Q11	?	X	X	.	?	.	?	?	X	.	.
	Q12	?	?	X	.	?	.	?	?	?	?	.
On whole strategy	Q13	.	?	X	.	?	.	?	?	?	X	.
	Q14	?	N	N	N	N	.	N	.	N	?	N
Score per participant		4.00	0.46	0.68	4.79	4.39	5.29	3.31	3.53	2.59	3.27	4.47

## 7 Discussion

From the responses to the questionnaire, and doubts expressed during the study, it became clear that the participants' main problem was to interpret the ranges of velocities represented by the animations. In addition, they frequently got confused by the short executions of the strategy, what made evident their low confidence on their interpretation of the presentation of the strategy. This results suggest three means of facilitating a better understanding of the graphical presentation of a strategy, all of them implemented in the current system: (1) provision of complementary explanations in natural language; (2) rewriting of the printed material; and (3) making explicit the action being executed at each moment in the short executions of the learner model.

<sup>4</sup> The probabilities of right ( $p_r$ ) and wrong ( $p_w$ ) answers were estimated (using Laplace estimate) and weights calculated as  $\omega_r = p_w$  and  $\omega_w = p_r$  (doubtful answers were considered half-right answers).

Despite the limitations of the study (e.g. size, subjective evaluation of correctness), the results obtained suggest that learners are able to achieve a reasonable understanding of their models. Some obstacles to comprehensibility of the models exist, however:

- Sets of rules with no explicit rationale (e.g. in terms of goals) may be inherently difficult to comprehend, no matter how small they are—this relates to the *opacity* of sets of rules discussed by Barr and Feigenbaum [3].
- The refined version in Fig. 3 of the model presented in Fig. 2 contains the same number of rules, but unfortunately that is not always the case. In fact, although half of the refined versions of the models obtained in our empirical study contain no more rules than the original models, a couple of refined versions were up to 1.6 times bigger.
- Graphical presentations in this context are more natural than numerical presentations, but they can be too concrete and less intuitive than correspondent expressions in natural language. The explanations produced by our system are, however, no more than simple verbalisations of rules preconditions and actions.
- Presenting ranges of angles by arcs, and ranges of positions by boxes appear to be straightforward choices. In contrast, the clarity of presenting ranges of velocities using animations is less obvious.

## 8 Conclusions

This case study has focused on producing learner models that are designed to be studied, and understood, by the learners. A number of reasons justify why the models so constructed should be easy to comprehend by learners: they are presented in a consistent modality, profiting from previous experience in the domain task; they are modular, allowing understanding of each component independently of others; and they can be reviewed and edited interactively, in an attempt to make their presentation more enticing. Qualitative verbalisations of the rules and short executions of the model focused on any rule are also provided. Each learner model is supported by data, which is displayed graphically in order to justify the existence and relevance of every component of the learner model, making the accountability of the models more apparent to learners, with the aim of increasing their acceptance of the models.

The results of an informative study suggest that learners can understand the graphical interface to learner models. The study also exhibited a number of factors that can affect the comprehensibility of the models and its graphical presentation. Despite the fact that further research is necessary to fully appraise the suitability of our approach, we are confident this paper gives convincing support for the claim that learners can comprehend a carefully designed presentation of their learner models, at least in the context of a sensorimotor control task.

## References

- [1] Anderson J.R.: Rules of the Mind. Lawrence Erlbaum Associates (1993)

- [2] Ayala G., Yano Y.: Learner models for supporting awareness and collaboration in a CSCL environment. In: Frasson C., Gauthier G. (eds.), *Intelligent Tutoring Systems (ITS'96)*, vol. 1086 of *Lecture Notes in Computer Science*. Springer-Verlag (1996) 158–167
- [3] Barr A., Feigenbaum E.A. (eds.): *The Handbook of Artificial Intelligence*, vol. 1. Pitman, London (1981)
- [4] Barto A.G., Sutton R.S., Anderson C.W.: Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics SMC-13* (1983) 834–846
- [5] de Buen P.R., Vadera S., Morales E.F.: A collaborative approach to user modeling within a multi-functional architecture. In: Kay [17] (1999) 291–293
- [6] Bull S.: See yourself write: A simple student model to make students think. In: Jameson A., Paris C., Tasso C. (eds.), *User Modeling (UM'97)*. Springer Wien New York (1997) 315–326
- [7] Bull S., Brna P., Pain H.: Extending the scope of the student model. *User Modeling and User-Adapted Interaction* **5** (1995) 45–65
- [8] Bull S., Pain H.: “Did I say what I think I said, and do you agree with me?” inspecting and questioning the student model. In: Greer [14] (1995) 501–508
- [9] Cohen W.W.: Fast effective rule induction. In: Prieditis A., Russell S. (eds.), *Machine Learning (ML'95)*. Morgan Kaufmann (1995)
- [10] Conlon T.: Alternatives to rules for knowledge-based modelling. *Instructional Science* **27** (1999) 117–146
- [11] Cook R., Kay J.: The justified user model: A viewable, explained user model. In: *Proceedings of the Fourth International Conference on User Modeling*. Hyannis, MA (1994) 145–150
- [12] Cox R., Brna P.: Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education* **6** (1995) 239–302
- [13] Dimitrova V., Self J., Brna P.: The interactive maintenance of open learner models. In: Lajoie S.P., Vivet M. (eds.), *Artificial Intelligence in Education (AIED'99)*. IOS Press (1999) 405–412
- [14] Greer J.E. (ed.): *Proceedings of the Seventh World Conference on Artificial Intelligence in Education*. Washington, DC (1995)
- [15] Greer J.E., Zapata-Rivera J.D., Ong-Scutchings C., Cooke J.E.: Visualization of Bayesian learner models. In: Morales R., Pain H., Bull S., Kay J. (eds.), *Proceedings of the Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling*. Le Mans, France (1999) 9–13
- [16] Hayes-Roth F.: Rule-based systems. *Communications of ACM* **28** (1985) 921–932
- [17] Kay J. (ed.): *User Modeling (UM'99)*. Springer Wien New York (1999)
- [18] Michie D., Chambers R.A.: BOXES: An experiment in adaptive control. In: Dale E., Michie D. (eds.), *Machine Intelligence*, vol. 2. Oliver and Boyd, Edinburgh (1968) 137–152
- [19] Mitchell T.M.: *Machine Learning*. McGraw-Hill (1997)
- [20] Morales R., Pain H.: Modelling of novices' control skills with machine learning. In: Kay [17] (1999) 159–168
- [21] Paiva A., Self J.A., Hartley R.: Externalising learner models. In: Greer [14] (1995)
- [22] Pineda L., Garza G.: A model for multimodal representation and inference. In: Paton R., Neilsen I. (eds.), *Visual Representations and Interpretations*. Springer (1999)
- [23] Self J.A.: Bypassing the intractable problem of student modelling. In: *Proceedings of ITS'88*. Montreal, Canada (1988) 18–24
- [24] Sison R., Shimura M.: Student modeling and machine learning. *International Journal of Artificial Intelligence in Education* **9** (1998) 128–158